Uvod u HTML

C201

priručnik za polaznike 2016 Srce

TEČAJEVI**srca**

srce
Sveučilište u Zagrebu
Sveučilišni računski centar

Ovu inačicu priručnika izradio je autorski tim Srca u sastavu:

Autorica: Gorana Kurtović

Recenzent: Nino Katić

Urednica: Irena Jandrić

Lektorica: Mia Kožul

TEČAJEVI srca

Sveučilište u Zagrebu

Sveučilišni računski centar

Josipa Marohnića 5, 10000 Zagreb

edu@srce.hr

ISBN 978-953-7138-94-3 (meki uvez)
ISBN 978-953-7138-95-0 (PDF)

Verzija priručnika: C201-20170223

srce otvoreni pristup

Sadržaj

Uvod

U zadnjih su dvadesetak godina *web*-stranice postale sveprisutne i jedan su od glavnih načina digitalne komunikacije. Za razliku od standardnih medija, *web* nudi mogućnost objave sadržaja i to na jednostavan način.

Cilj je ovog tečaja upoznati polaznika s osnovama HTML-a – tehnologije za izradu *web*-stranica. Preduvjet za pohađanje i razumijevanje ovog tečaja jest poznavanje osnova rada na računalu.

Iako priručnik nastoji biti iscrpan, određeni broj rjeđe korištenih ili vrlo naprednih svojstava i mogućnosti HTML-a nije mogao biti uvršten u tečaj.

Priručnik se sastoji od uvoda, 12 poglavlja i pet praktičnih vježbi za čije su izvođenje potrebni dodatni materijali.

U priručniku se rabe ovi simboli:

- kurzivom su istaknuti nazivi HTML-elemenata te CSS-svojstava, ključnih riječi i funkcija
- podebljano su označeni važni pojmovi te naziv HTML-svojstva kad se navodi prvi put
- u okvirima sa strane navedene su zanimljivosti i napomene.

Na kraju svakog poglavlja nalazi se pregled obrađenih cjelina.

1. *Web*-stranice

Po završetku ovog poglavlja polaznik će moći:

- *objasniti što je to HTML*
- *objasniti što je web-stranica i gdje se web-stranice nalaze*
- *osmisliti informacijsku strukturu jednostavne web-stranice*
- *objasniti pojmove SEO, pristupačnost i prilagodljivost web-stranica*

Web-stranice se izrađuju uporabom HTML jezika pomoću kojeg se definiraju struktura i sadržaj. Uz CSS, kojim se *web*-stranice oblikuju i pomoću kojeg poprimaju svoj izgled, HTML je osnovna tehnologija na kojoj se temelji izrada *web*-stranica.

1.1. O HTML-u i izradi *web*-stranica

Osnovni gradivni element svake *web*-stranice je HTML-kôd.

HTML je kratica za *HyperText Markup Language*. HTML je jezik za označavanje kojim se određuje struktura, sadržaj i funkcija nekog HTML-dokumenta (*web*-stranice). Pomoću HTML-a određuju se važni elementi svakog HTML-dokumenta kao što su npr. naslov, odlomak, slika, poveznica/veza (engl. *hyperlink*) i sl.

Dakle, HTML je jezik za označavanje pomoću kojega se može odrediti struktura elemenata unutar HTML-dokumenta. Osim strukture elemenata u dokumentu definirane HTML-om, CSS omogućava da se ti elementi stilski/grafički urede. Na taj se način *web*-pregledniku daje do znanja kako će stranica izgledati prilikom prikaza.

Zanimljivosti i napomene

CSS-kôd se može dodati u HTML dokument, ali ispravnije je CSS izdvojiti u zaseban dokument, povezan s HTML-dokumentom.

Razvoj HTML-a

HTML je početkom 90-ih godina prošlog stoljeća razvio Tim Berners-Lee, fizičar zaposlen u CERN-u. HTML je vrlo brzo stekao veliku popularnost koja je tijekom 90-ih utjecala na nagli razvoj Interneta. Zbog tako naglog razvoja izašlo je nekoliko HTML standarda koji su međusobno bili neusklađeni. Od pojave HTML-a do prve prave uporabljive inačice, koju je izdala **W3C** organizacija (engl. *World Wide Web Consortium*), prošlo je šest godina.

HTML je u početku omogućavao jednostavno strukturirane web-stranice - odlomke, prijelome redova i zaglavlja, a nije omogućavao unos grafike i multimedijalnih elemenata. HTML se razvijao, a da se pri tome nije pridavala pažnja oblikovanju i prezentaciji stranice, odnosno izgledu, jer je osnovni cilj bio strukturiranje podataka na jednostavan način. Međutim, pojavom vizualnih internetskih preglednika, a zbog brzog i nekontroliranog razvoja pojavili su se problemi oko međusobne nepodrživosti različitih preglednika na različitim platformama. Ti su problemi aktualni i danas, ali su sve manje izraženi.

Svaka nova verzija HTML-a donosila je novosti i unapređenja te je predstavljala napredak. Ovaj napredak (većinom) prate i internetski preglednici pa se u najnovijim preglednicima HTML kôd ispravno prikazuje. Problem se javlja zbog toga što korisnici često nemaju instalirane najnovije verzije preglednika. Ovaj se problem rješava na način da se prilikom pisanja HTML kôda pazi da on bude kompatibilan s nekoliko zadnjih inačica internetskih preglednika.

Aktualna inačica HTML-a zove se HTML5 i podržana je u svim novijim internetskim preglednicima.

Učenjem novih inačica tehnologija na kojima se temelji izrada *web*-stranica (HTML5, ali i CSS3) polaznici ovog tečaja bit će u stanju razumjeti i starije inačice.

Kako korisnici pristupaju *web*-stranicama

Web-stranicama pristupa se korištenjem ***web*-preglednika** (eng. *web-browser*), kao što su *Firefox, Chrome, IE, Safari, Opera, Microsoft Edge,* itd. Da bi se neka *web*-stranica učitala u preglednik potrebno je u njega upisati *web*-adresu. Pod uvjetom da je korisnik spojen na Internet, preglednik šalje zahtjev *web*-poslužitelju, računalu na kojem se tražena stranica nalazi. Poslužitelj šalje pregledniku traženu stranicu, a preglednik zatim stranicu prikazuje korisniku.

***Web*-poslužitelj** (*web server*) je računalo na kojem se nalaze *web*-stranice, koje se nalazi bilo gdje u svijetu i koje je uvijek spojeno na Internet kako bi moglo odgovoriti slanjem *web*-stranice korisniku u bilo kojem trenutku. Iako neke veće tvrtke imaju dedicirane *web*-poslužitelje na kojima se nalaze njihove stranice, većina će korisnika prilikom izrade i smještanja vlastitih *web*-stranica ipak zakupiti prostor kod neke tvrtke koja ima uslugu "udomljavanja" *web*-stranica, tj. *hosting* (engl. *web hosting*).

Zanimljivosti i napomene

Neki se korisnici za pristupanje *web*-stranicama koriste i **čitačima** (eng. *screen readers*), uređajima koji izgovaraju sadržaj *web*-stranica. O ovim će korisnicima i čitačima biti riječi kasnije.

Postoji veliki broj različitih **uređaja** s kojih se može pristupiti *web*-stranicama kao što su stolno računalo, laptop, tablet i mobitel. Važno je imati na umu da svi ovi uređaji imaju različitu veličinu ekrana tj. rezoluciju te je pri izradi *web*-stranica potrebno paziti kako će se one prikazivati na različitim uređajima.

Izrada *web*-stranica

U ranijim fazama razvoja *weba* najvažnije je bilo znati napisati HTML kôd i objaviti *web*-stranicu. Ipak, danas je izrada *web*-stranica postala puno više od samog pisanja HTML i CSS-kôda.

Povećanjem dostupnosti i pojeftinjenjem tehnologije i internetske veze te razvojem mobilnih tehnologija *web* je postao dostupan većem broju korisnika. Ovo je dovelo do povećanja broja *web*-stranica iste ili slične tematike koje su dostupne puno većem broju korisnika nego prije.

Zbog povećanja konkurencije sada je potrebno uložiti i puno više truda pri razvoju *web*-stranica kako bi korisnici došli baš do vaše stranice. Potrebno je pažljivo osmisliti koncepciju *web*-stranice, sadržaj, prezentaciju, jednostavnost snalaženja na stranici, prilagođenost stranice za različite uređaje ili različite kategorije korisnika i pronalažljivost *web*-stranica kako bi poruka koju vlasnik *web*-stranice želi prenijeti došla do krajnjeg korisnika.

Naglasak se pri izradi posebno stavlja na korisnika. Potrebno je jasno odlučiti kome je *web*-stranica namijenjena, zašto korisnici na nju dolaze i što žele postići, koje im informacije trebaju, koliko često posjećuju stranicu, gdje se zadržavaju i slično.

Manje se stranice, statičnijeg sadržaja još uvijek izrađuju pisanjem HTML i CSS-kôda. Veće stranice, pogotovo one koje se osvježavaju često, češće se izrađuju uz pomoć sustava za upravljanje *web*-sadržajem, tj. CMS-a (engl. *Content management system*). CMS omogućuje osobi koja izrađuje *web*-stranice korištenje nekih naprednijih opcija i *web*-tehnologija s manje znanja te jednostavnije održavanje sadržaja bez svakodnevnog susretanja s HTML-om i CSS-om. Međutim, znanje HTML-a (i CSS-a) uvelike pomaže čak i kod rada sa sustavima za upravljanje sadržajem jer omogućuje bolje razumijevanje pozadinskih procesa te bolju kontrolu nad izgledom *web*-stranica. Za izradu *web*-stranice u CMS-u (uglavnom) nije potrebno poznavanje programskih jezika i drugih *web*-tehnologija.

1.2. Organizacija *web*-stranica (informacijska struktura)

Jednom kad je poznata ciljana skupina korisnika *web*-stranice može se početi razmišljati o organizaciji *web*-stranice ili o njezinoj **informacijskoj strukturi**.

Informacije se prvo trebaju organizirati u sekcije i stranice. Cilj ovoga procesa jest izrada dijagrama stranica koji će se koristiti u strukturiranju *web*-stranice. Na kraju procesa trebala bi nastati mapa *web*-stranica (*site map*) koja prikazuje strukturu, organizaciju i grupiranje stranica.

Tehnika sortiranja kartica može biti dobra pomoć u organizaciji izrade mape. Tehnika se sastoji od toga da se na različite papiriće zapiše svaka pojedinačna informacija koja bi korisniku mogla biti korisna te se zatim kartice grupiraju u stranice ili u grupe stranica. Ponekad je u ovoj

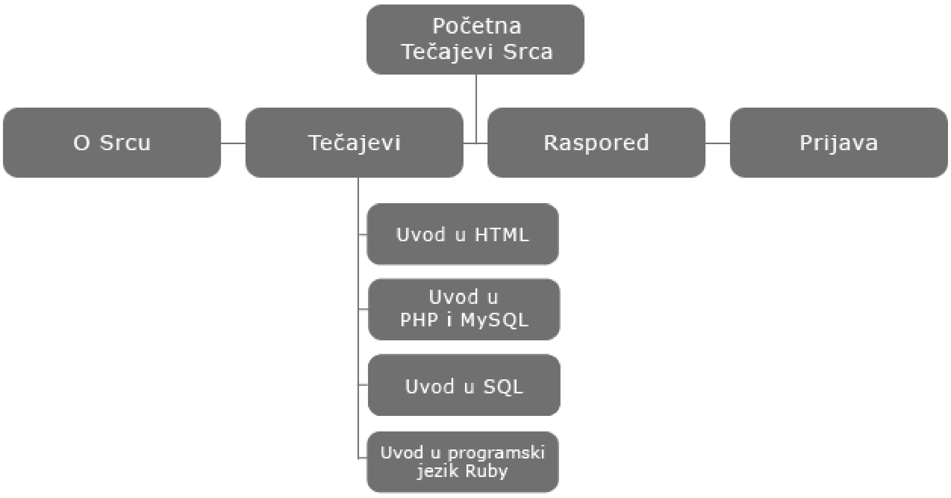
Zanimljivosti i napomene

Web-stranica (engl. *web page*) označava samo jednu stranicu koja se od početka do kraja može pregledati bez dodatnog klikanja na *linkove*. Riječ je o jednom HTML-dokumentu.
Web-sjedište (engl. *web site*) označava određeni broj tematski ili drugačije vezanih stranica. Budući da je riječ o većem broju stranica, ponekad se za *web*-sjedište jednostavno koristi plural *web*-stranice. *Web*-sjedište sastoji se od više povezanih HTML-dokumenata.

fazi korisno konzultirati se s korisnicima kako bi se informacije lakše grupirale.

Uobičajeno svi *webovi* kao početnu točku imaju početnu stranicu - naslovnu (*homepage*). Ako je riječ o kompleksnom *webu* moguće je imati i više početnih stranica, svaku za jednu temu.

Mapa *weba* Tečajeva Srca izgledala bi npr. ovako:



Osim organizacije stranica i sekcija važno je napomenuti da se i sadržaj svake pojedine stranice treba organizirati vizualno. Većina korisnika ne čita cijele *web*-stranice nego ih samo brzinski pregleda u potrazi za traženim informacijama. Vizualna hijerarhija se odnosi na put kojim oči korisnika prolaze i doživljavaju ono što vide, a može se kontrolirati povećavanjem kontrasta određenih područja za koja smatramo da su korisniku od interesa.

Pažnju korisnika tako će uvijek privući veća slova, svijetlija područja, slike i kontrasti u tekstu (npr. podebljana slova). Dodatni naglasak na sadržaj postiže se i vizualnim grupiranjem pojedinih informacija na stranici.

1.3. Optimizacija *web*-stranica za tražilice (SEO)

Zanimljivosti i napomene

SEO je iznimno kompleksno i opsežno područje te se ne može detaljno obuhvatiti u ovome tečaju. Ako želite saznati nešto više o optimizaciji za tražilice pročitajte knjigu Razvoj pronalažljivih Web stranica, autora Aarrona Waltera.

Dobra **optimizacija za tražilice** (pronalažljivost ili SEO – *Search engine optimization*) često može biti presudna kada je u pitanju dolazak korisnika na *web*-stranice s neke tražilice. SEO služi tome da bi se prilikom pretrage određenih pojmova u tražilici, *web*-stranica pojavila što bliže prvom rezultatu.

Kako bi se *web*-stranica optimizirala za tražilice potrebno je koristiti nekoliko tehnika. Jedna od njih je zaključiti koje pojmove korisnici zapravo traže te pametnim smještanjem tih pojmova na određenim mjestima na *web*-stranici povećati mogućnost da tražilica vrati link na našu *web*-stranicu.

Postoji nekoliko mjesta na koja se mogu smjestiti ključne riječi (za koje mislimo da će ih korisnici pretraživati) kako bi se povećala pronalažljivost *web*-stranice:

- Naslov stranice - `<title>`element
- URL/adresa stranice – gdje je moguće trebale bi se koristiti ključne riječi prilikom imenovanja dokumenata
- Naslovi tekstova – ako se ključne riječi smještaju u naslove tj. unutar `<h>` elemenata, tražilica će znati da je taj pojam važan za *web*-stranicu
- Tekst – ako je prihvatljivo, ključna riječ bi se trebala pojavljivati nekoliko puta u tekstu
- Tekst poveznice/*linka* – ključne riječi bi trebale biti i dio *linka* na druge stranice gdje je moguće, umjesto riječi kao što su *više* i *na ovoj stranici*
- Alternativni tekst za sliku (ALT) – svakoj slici bi se trebao staviti `<alt>` atribut koji opisuje sliku
- Opis stranice – `<meta>` element

Tražilice ne uzimaju u obzir samo ključne riječi nego i broj linkova s drugih stranica na našu *web*-stranicu. Važnijim se smatraju linkovi sa stranica koje imaju sličan sadržaj kao i naša. U slučaju da se unutar `<a>` elementa nalaze i ključne riječi od interesa, taj će se link smatrati još važnijim.

Npr. ako imamo stranicu o školi koja održava informatičke tečajeve, relevantniji će biti link koji je na stranici neke druge škole, nego npr. na stranici frizerskog salona. Još je važniji ako na njemu umjesto same adrese *web*-stranice ili riječi "više" piše tekst "informatički tečajevi".

Za uspješan SEO od iznimne je važnosti uspješno određivanje ključnih riječi.

1.4. Pristupačnost

Čini se gotovo nepotrebnim spominjati **pristupačnost** informacijama na *web*-stranicama, ali potrebno je uzeti u obzir da postoje ljudi kojima je nešto teže koristiti računala, Internet pa i pregledavati *web*-stranice. Riječ je o osobama s invaliditetom. Cilj svake dobre *web*-stranice trebao bi biti omogućiti svim korisnicima jednaku mogućnost pristupanja informacijama, neovisno o njihovim mogućnostima i uređajima kojima se koriste.

Postoji više kategorija osoba s invaliditetom na koje bi trebalo misliti prilikom izrade *web*-stranica, ali najteže je prilagoditi stranice slijepim i slabovidnim osobama.

Zanimljivosti i napomene

Detaljnije informacije o prilagodbi *weba* osobama s invaliditetom moguće je pronaći na specijaliziranim stranicama organizacija koje se bave ovom temom; WAI, WCAG, webaim… te na stranici http://www.srce.unizg.hr/kontakt/pristupacnost/#smjernice i http://www.eduquality-hr.com/index.php?option=com_content&view=section&layout=blog&id=9&Itemid=124.

Slijepe osobe često koriste čitače kako bi pregledavale *web*-stranice. Čitači su uređaji koji, kao što i samo ime kaže, čitaju sve informacije s *web*-stranica na glas. Slijepe osobe ne mogu preletjeti sadržaj i početi čitati ono što ih zanima, nego čitači za njih prolaze sav sadržaj dok ne dođu do onoga što korisnika zanima. Kako bi se omogućilo preskakanje sadržaja važno je prilikom izrade *web*-stranica koristiti HTML5 i semantičke oznake. Osim ovoga, vrlo je važno svim slikama i linkovima postaviti jasne i jednoznačne alternativne opise te paziti da tekst linkova ima jasno značenje (izbjegavati linkove sadržaja "više", "na stranici" i sl.).

Slabovidnim osobama važna je mogućnost povećanja i smanjenja slova na stranici, općenita čitljivost teksta (veća slova, jasno odvojeni tekstovi i velike margine između elemenata na stranici, veći proredi, povećani kontrasti između teksta i pozadine…). Osim ovoga, preporuča se i izrada dodatnog vizualnog predloška s povećanim kontrastom (npr. crna pozadina s bijelim slovima ili plava pozadina sa žutim slovima).

Za neke kategorije osoba s invaliditetom, npr. za osobe s poremećajem koncentracije, iznimno je važno poravnati sav tekst lijevo (nikako obostrano) te imati što jasniju, jednostavniju i dosljednu navigaciju.

Svim osobama s invaliditetom važno je da pred sobom imaju jasne, dobro i jednostavno napisane, jednostavne, razumljive i sažete *web*-stranice izrađene s mišlju o jednostavnosti korištenja, tj. korisnosti stranica (engl. *usability*).

1.5. Prilagodljivost za različite uređaje

Kao što je već rečeno, postoji veliki broj različitih **uređaja** s kojih korisnik može pristupiti *web*-stranicama kao što su stolno računalo, laptop, tablet i mobitel. Svi ovi uređaji imaju različitu veličinu ekrana tj. rezoluciju te je potrebno paziti kako se *web*-stranice prikazuju na različitim uređajima.

Responzivni *web*-dizajn (RWD) pojam je koji je već postao uobičajen, a odnosi se na izradu *web*-stranica koja automatski osigurava optimalno pregledavanje *web*-stranica na različitim uređajima. Ovime se omogućava dobra čitljivost, dobra i jasna navigacija po stranicama bez potrebe za povećavanjem, smanjivanjem ili micanjem stranice lijevo ili desno kako bi se vidio cijeli sadržaj.

Ideja iza responzivnog *web*-dizajna jest da se na svim uređajima prikazuje isti sadržaj, ali da se, ovisno o uređaju na kojem se stranica gleda, izgled stranice prilagođava na način da sadržaj ostane jasan i čitljiv.

Obično je riječ o nekoliko inačica istog dizajna koji se neprimjetno mijenja ovisno o veličini ekrana na kojem korisnik pregledava *web*-stranice. Responzivnost se uglavnom postiže izradom posebnih CSS uvjeta za različite veličine ekrana.

Neke stranice nude popis najčešće korištenih uređaja i veličina ekrana, tj. rezolucija kako bi osobe koje izrađuju *web* lakše znale za koje veličine trebaju prilagoditi dizajn.

Na stranici http://www.w3schools.com/browsers/browsers_display.asp može se vidjeti popis najčešće korištenih veličina ekrana.

Screen Resolution Statistics

As of today, about 97% of our visitors have a screen resolution of 1024x768 pixels or higher:

Date	Other high	1920x1080	1366x768	1280x1024	1280x800	1024x768	800x600	Lower
January 2015	32.7%	16%	33%	7%	5%	4%	0.3%	2%
January 2014	34%	13%	31%	8%	7%	6%	0.5%	0.5%
January 2013	36%	11%	25%	10%	8%	9%	0.5%	0.5%
January 2012	35%	8%	19%	12%	11%	13%	1%	1%
January 2011	50%	6%		15%	14%	14%	0%	1%
January 2010	39%	2%		18%	17%	20%	1%	3%
January 2009	57%					36%	4%	3%

Dodatni je problem što osim veličine ekrana postoje još i različite orijentacije (portret ili pejzaž) te ekrani s različitim omjerima (3:4, 16:9…) što može biti problem kada se izrađuje stranica koja u svojoj pozadini ima sliku.

2. Osnove rada s HTML-om

Po završetku ovog poglavlja polaznik će moći:

- *objasniti od čega se sastoji HTML-element*
- *objasniti čemu služe HTML-atributi*
- *izraditi HTML-dokument s osnovnim elementima*
- *postaviti kodnu stranicu u HTML-dokumentu.*

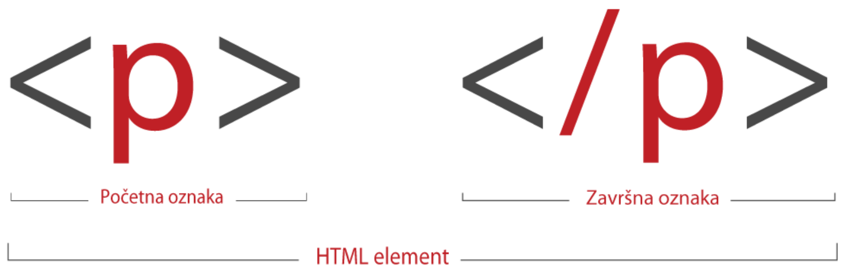
Kao i bilo koji drugi dokument, HTML-dokument ima određenu strukturu. HTML-dokument tako može imati naslov, odlomak, slike, tablice i slično.

U nastavku se nalazi vrlo jednostavan primjer HTML-dokumenta koji ima naslov i odlomak. HTML-elementi označeni su crvenom bojom.

```
<html>
<body>
       <h1>Tečajevi Srca</h1>
       <p>Sveučilišni računski centar više od
       četrdeset godina održava stručne tečajeve iz
       područja informacijskih i komunikacijskih
       tehnologija (ICT).</p>
</body>
</html>
```

2.1. Anatomija HTML-elementa

HTML-element sastoji se od početne i završne **HTML-oznake** (*tag*).



Sadržaj se unosi između dvije HTML-oznake te se one na neki način ponašaju poput spremnika koji pojašnjavaju koja vrsta informacije se nalazi u tom spremniku. Na primjer, unutar HTML-oznaka `<p> i </p>` nalazit će se jedan odlomak teksta.

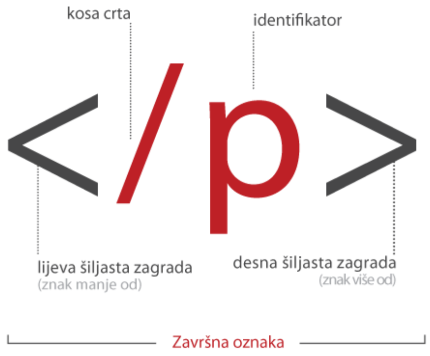
Kako bi do kraja razjasnili anatomiju HTML-elementa potrebno ga je raščlaniti na osnovne dijelove. Svaka početna HTML-oznaka sastoji se od šiljastih zagrada i identifikatora.



Šiljaste zagrade karakteristične su za svaki HTML-element, tj. svaka HTML-oznaka mora imati dvije šiljaste zagrade.

Identifikator određuje o kojem se HTML-elementu radi.

Završni HTML-element sastoji se od šiljastih zagrada unutar kojih se nalazi kosa crta i identifikator. Kosa crta je oznaka koja pokazuje da je riječ o završnom elementu, a ne početnom.



Zanimljivosti i napomene

Do nedavno su se svi elementi (i oni koji imaju po dvije oznake, i oni koji imaju samo jednu) obavezno trebali zatvarati jer je preporuka bila pisati kôd prema XHTML standardu.No, s dolaskom HTML5 standarda elementi koji imaju samo jednu oznaku više se ne trebaju zatvarati iako je to i dalje moguće ako *web* treba proći strožu validaciju. Budući da je riječ o samo jednoj oznaci koja čini cijeli HTML-element, kosa crta stavlja se iza identifikatora, a ne ispred pa se u tom slučaju piše `<hr />` i `<br />`.

Na primjeru HTML-kôda koji se sastoji samo od jednog odlomka teksta može se vidjeti kako izgleda ispravno napisan odlomak teksta u HTML-u.

Primjer:

```
<p>
Sveučilišni računski centar više od četrdeset godina
održava stručne tečajeve iz područja informacijskih i
komunikacijskih tehnologija (ICT).
</p>
```

Kako bi kôd bio čitljiviji, može ga se pisati na način da svaki HTML-element bude u svom redu.

Postoje elementi poput naslova <h1> ili odlomka <p>, koji u sebi mogu imati sadržaj, ali postoje i elementi koji nemaju nikakav sadržaj, poput horizontalne linije koja se proteže preko cijele stranice i predstavlja tematski prijelom između odlomaka ili oznake za prelazak u novi red.

Na prethodnom primjeru može se vidjeti da se element za odlomak otvara na način da se ispred sadržaja napiše `<p>`, zatim slijedi tekst nakon čega se element naslova zatvara s `</p>`.

HTML-elementi koji nemaju sadržaj imaju samo jednu HTML-oznaku. Tako HTML-element za vodoravnu crtu, s obzirom na to da nema sadržaja ima samo jednu oznaku `<hr>`, dok HTML-element za prelazak u novi red ima samo oznaku `<br>`.

2.2. HTML atributi

Neki HTML-elementi mogu imati dodatna svojstva unutar početne HTML-oznake koja daju više informacija o samom sadržaju elementa. Uglavnom se sastoje od dva dijela: imena i vrijednosti, odvojenih znakom jednakosti. Ta se svojstva nazivaju **HTML-atributima**. Jedan HTML-element može imati i više atributa.

Ime atributa podatak je koji govori o kojoj vrsti informacije je riječ u atributu, dok je vrijednost informacija ili svojstvo atributa. Ime atributa trebalo bi se pisati malim slovima, a vrijednost atributa uvijek se mora nalaziti u dvostrukim navodnicima.

Većina atributa koristi se s točno određenim HTML-elementom te nije obavezna. Ipak, postoje HTML-elementi koji moraju imati i atribut, jer u suprotnom nije moguća pravilna interpretacija elementa. Primjer neizostavnog atributa je `src` (*source*, koristi se za označavanje putanje do slike).

HTML-element sa skupom atributa može se vidjeti na primjeru elementa za umetanje slike `<img>`. U ovome primjeru oznaka ima tri atributa.

Primjer:

```
<img src="Slike/Srce.jpg" height="55" width="65">
```

Atribut	Ime atributa	Vrijednost atributa
`src="Slike/Srce.jpg"`	`src`	`Slike/Srce.jpg`
`height="55"`	`height`	`55`
`width="65"`	`width`	`65`

Kod nekih atributa vrijednost se uopće ne navodi. Radi se o atributima koji imaju samo dvije vrijednosti – istina i laž, odnosno atribut je naveden ili atribut nije naveden. Ovakvi atributi nazivaju se još i Booleovim atributima.

U sljedećem primjeru prisutnost atributa `checked` naznačuje da će kvačica prvotno biti prikazana kao označena.

Primjer:

```
<input type="checkbox" checked>
Da, želim primati obavijesti.
```



Ako se želi da kvačica prvotno bude isključena, potrebno je izostaviti atribut `checked`.

Primjer:

```
<input type="checkbox">
Da, želim primati obavijesti.
```



Određene atribute (tzv. globalne atribute) moguće je postaviti na bilo koji HTML-element. Objašnjenje najčešće korištenih globalnih atributa dano je u sljedećoj tablici:

Atribut	Objašnjenje
`Id`	Identifikator elementa koji se koristi za referenciranje elementa u CSS-u i JavaScriptu. Treba biti jedinstven na razini stranice.
`class`	Klasa elementa koja se koristi za referenciranje elementa u CSS-u i JavaScriptu. Na stranici može biti više elemenata koji imaju istu klasu.
`Title`	Dodatne informacije o elementu. Najčešće se radi o opisu elementa koji će biti prikazan kad korisnik prijeđe mišem preko elementa (u *tooltipu*).
`style`	Atribut preko kojega je moguće postaviti *inline* CSS-pravila na neki element.
`Lang`	Jezik elementa. Najčešće se postavlja na razini cijele stranice tako da se postavi na elementu HTML, ali moguće ga je postaviti i na drugim elementima. Vrijednosti su oznake jezika npr. `hr, en, de`

2.3. DOCTYPE deklaracija

DOCTYPE deklaracija opisuje koja se inačica HTML-a koristi u dokumentu. Potrebno ju je navesti na samom početku HTML dokumenta, prije svih drugih elemenata. DOCTYPE deklaracija nije HTML-element, nego instrukcija pregledniku, kako bi preglednik znao koju inačicu HTML-a treba interpretirati.

Zanimljivosti i napomene

DOCTYPE deklaracije za starije inačice HTML-a bile su dosta kompleksnije pa je npr. za inačicu HTML 4.01 Strict, DOCTYPE deklaracija izgledala ovako:

```
<!DOCTYPE HTML
PUBLIC "-//W3C//DTD
HTML 4.01//EN"
"http://www.w3.org/T
R/html4/strict.dtd">
```

Zanimljivosti i napomene

HTML-kôd može se pisati u različitim programima, pa tako i u programu **Blok za pisanje** (engl. *Notepad*). Za potrebe ovog tečaja koristi se besplatni program **Brackets** koji je na računalima već instaliran.

Program je dostupan na adresi http://brackets.io/, a potrebno ga je preuzeti preko **poveznice Download Brackets without Extract** ispod velikog plavog dugmeta (besplatna inačica programa).

S dolaskom inačice HTML5 DOCTYPE deklaracija je postala znatno jednostavnija te je dovoljno na početku dokumenta napisati ovo:

```
<!DOCTYPE html>
```

2.4. Osnovna struktura HTML-dokumenta – *html, head, title i body*

Svi HTML dokumenti, neovisno o kompleksnosti i sadržaju imaju neke zajedničke dijelove kojima se definira struktura HTML-dokumenta. Elementi koji su zajednički svim HTML-dokumentima su:

- `<html></html>`
- `<head></head>`
- `<title></title>`
- `<body></body>`

Element koji je obavezan na svakoj stranici je `html`. To je prvi element, unutar kojeg se nalazi sav sadržaj nekog HTML-dokumenta, a služi kako bi *web*-preglednik znao prepoznati da je riječ o HTML-dokumentu. Budući da je sav sadržaj dokumenta unutar ovog elementa, cijeli dokument počinje i završava oznakama `<html>` i `</html>`.

Primjer:

```
<html>
<head>
<title>Mudre poslovice</title>
</head>
<body>
</body>
</html>
```

Na vrhu svakog HTML-dokumenta prvo se navodi HTML-element `<head>` koji sadrži informacije o stranici, koje se ne prikazuju kao dio sadržaja stranice. Taj element predstavlja zaglavlje HTML-dokumenta.

U `<head>` elementu nalazi se element `<title>` u kojem se smješta naslov te *web*-stranice. Naslov stranice prikazuje se na naslovnoj traci preglednika i ne mora biti identičan naslovu HTML-dokumenta koji se pohranjuje lokalno na računalu.

Sav je sadržaj koji se nalazi unutar elementa `<body>` sadržaj koji će se prikazati u internetskom pregledniku. U gornjem primjeru prikazat će se prazna stranica, bez sadržaja.

Ako se između oznaka `<body>` upiše neki tekst, on će postati sadržaj stranice i pregledom dokumenta u pregledniku taj će tekst biti vidljiv.

S obzirom na to da su HTML-dokumenti zapravo obične tekstualne datoteke koje se sastoje od HTML-kôda, preporuka je taj kôd pisati strukturirano, tj. **uvučeno** radi preglednosti, kako bi se kasnije u dokumentu lakše snalazili.

Zanimljivosti i napomene

Kôd bilo koje HTML-stranice, i one koju nismo sami izradili, može se vidjeti iz internetskog preglednika, npr. u programu **Mozilla Firefox** desnim klikom na stranicu pristupi se kontekstualnom izborniku i odabere **Izvor** (engl. *View →Source*).

Zanimljivosti i napomene

HTML-kôd se može pisati uvučeno zbog toga što *web*-preglednik ignorira dodatne razmake i prelaske u novi red, kad naiđe na njih.

Ovo je poznato pod nazivom „*white space collapsing*“, a autori HTML-dokumenata to koriste kako bi pisali pregledniji HTML-kôd.

HTML-dokument pohranjuje se, za početak, lokalno na računalu kao datoteka s ekstenzijom *.html*, npr. mudre-poslovice.html.

2.5. Informacije o stranici – *meta* element

Meta podaci su različiti dodatni podaci o *web*-stranici, koji krajnjem korisniku nisu vidljivi, a mogu imati različitu svrhu. Meta podaci mogu sadržavati popis ključnih riječi, opis stranice, informacije o autoru i slično koje *web*-stranica šalje tražilicama radi bolje pretraživosti i optimizacije.

Meta podaci se smještaju unutar elementa `<head>`.

Za označavanje meta podataka koristi se HTML-element `<meta>` koji nema završnu oznaku. Vrijednost meta podatka upisuje se u atribut.

Najčešći atributi su **name** ili **content** atributi koji se često koriste zajedno i označavaju svojstva cijele HTML-stranice. Vrijednost atributa *name* je svojstvo koje se postavlja, dok je vrijednost atributa *content* vrijednost koja se želi tom svojstvu dodijeliti, tj. specifikacija atributa *name*.

Vrijednost atributa *name* može se proizvoljno postaviti, ali najčešće se koriste:

- **content-type** – dodatno deklariranje kodne stranice.
- **description** – sadrži opis stranice do 155 znakova koji tražilicama olakšava razumijevanje sadržaja na stranici. Kao rezultat pretrage u tražilicama ispisuje se ime stranice i opis koji je na ovom mjestu postavljen.
- **keywords** – popis ključnih riječi odvojenih zarezom za koje se pretpostavlja da bi ih korisnik mogao koristiti za pretraživanje u tražilicama. Dok je ovo prije bio vrlo važan korak za SEO optimizaciju, danas sve više gubi na važnosti.
- **author** – definira se autor stranice.

Primjer korištenja meta podataka:

```
<html>
<head>
<title>Tečajevi Srca</title>
          <meta name="description" content="Popis
          osnovnih informatičkih tečajeva dostupnih
          u Sveučilišnom računskom centru">
</head>
<body>
</body>
</html>
```

2.6. Postavljanje kodne stranice

Kôdna stranica određuje koja se kombinacija nula i jedinica koristi za zapisivanje pojedinog znaka. Znakovi koji pripadaju engleskoj abecedi imaju iste kombinacije u svim kodnim stranicama, te se uvijek ispravno prikazuju, ali do problema dolazi kada je potrebno prikazati posebne znakove kojima se koriste drugi jezici.

Kako bi se hrvatski dijakritički znakovi ispravno prikazivali, potrebno je postaviti pravu kôdnu stranicu koja ih sadrži.

Najbolji izbor je kôdna stranica **UTF-8**, koja može prikazati gotovo sve svjetske nacionalne znakove.

Kôdna stranica se zadaje pomoću elementa `<meta>` kojem se postavlja atribut **`charset`**. Ovaj element dolazi unutar elementa `<head>`.

```
<head>
    ...
    <meta charset="utf-8">
</head>
```

Osim kôdne stranica UTF-8, za prikaz hrvatskih znakova često se koriste i kôdne stranice Windows-1250 i ISO-8859-2.

```
<head>
    ...
    <meta charset="windows-1250">
</head>
```

Ako se HTML-datoteka ne prikazuje ispravno, korisnik može i sam promijeniti kôdnu stranicu u pregledniku (pomoću opcije *View*→**Text Encoding** ili slično).

Zanimljivosti i napomene

Bitno je da sama HTML-datoteka također bude spremljena na računalo u istoj kôdnoj stranici.

2.7. Komentari

HTML-dokument može sadržavati i komentare koji omogućavaju unošenje sadržaja koji nije vidljiv krajnjem korisniku. Komentari služe kako bi osoba koja izrađuje stranicu imala mogućnost zabilježiti pojašnjenja vezana uz neki dio stranice ili kôda u samom HTML-dokumentu. Na taj se način olakšava naknadni rad na dokumentu.

Također, mogu služiti za obilježavanje dijelova kôda radi jednostavnije i brže preglednosti. Npr. komentarom se može označiti zaglavlje i podnožje *web*-stranice ili pojedini odlomak teksta. Krajnjem korisniku, komentari neće biti prikazani, a osobi koja izrađuje *web*-stranicu omogućit će lakši pronalazak onog dijela kôda u kojem želi napraviti intervenciju.

Osim ovoga, komentari mogu biti i prilično korisni u slučaju potrebe testiranja. Ako se želi vidjeti kako bi stranica izgledala bez nekog dijela sadržaja, dovoljno je prije tog sadržaja staviti početnu oznaku komentara, a nakon njega završnu oznaku komentara, kako bi sadržaj

koji ne želimo vidjeti bio skriven. Taj se dio sadržaja tada neće prikazati. Čim se oznake komentara obrišu, on će opet biti vidljiv.

Da bi se dio kôda sakrio ili pretvorio u komentare potrebno je dodati HTML-oznake za komentare. Početna oznaka je <!-- i dodaje se neposredno prije dijela kôda koji se želi pretvoriti u komentare, a završna -->i dodaje se neposredno poslije dijela kôda koji se želi pretvoriti u komentare, tj. sakriti od prikaza krajnjem korisniku.

Primjer:

```
<!-- Ovo su četiri stiha iz pjesme "Jesensko pismo"
Dragutina Tadijanovića, ispisana u jednom odlomku -->

<p>Prošla je berba, milo moje srce, I nema na trsju
grozdova. Na lozama se lišće trese kao ruke staraca.
Boje se tope, i nestaju.</p>
```

Prošla je berba, milo moje srce, I nema na trsju grozdova. Na lozama se lišće trese kao ruke staraca. Boje se tope, i nestaju.

3. Rad s tekstom

Po završetku ovog poglavlja polaznik će moći:

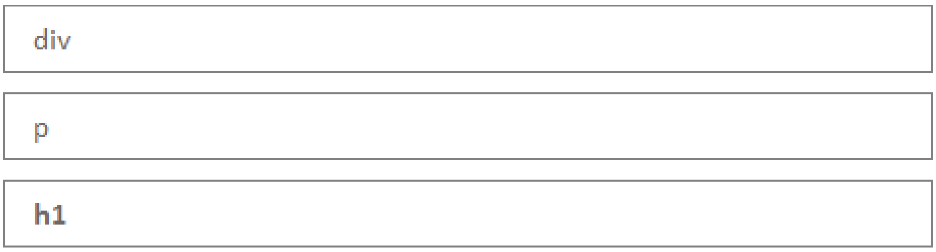
- *objasniti razliku između blok i inline elementa*
- *izraditi osnovni tekstualni dokument u HTML-u*
- *koristiti posebne znakove i predformatirani tekst*
- *razlikovati stilske i semantičke HTML-elemenate za oblikovanje teksta te njihovu upotrebu.*

HTML-dokumenti mogu se sastojati od teksta, slika, linkova, animacija i drugih sadržaja. U ovome poglavlju bit će objašnjeno kako se u HTML-dokumentu radi s tekstom, kako se taj tekst na kraju prikazuje u *web*-pregledniku te može li taj tekst nositi i neke dodatne informacije.

3.1. Blok i *inline* elementi

Prema načinu prikaza u HTML-u postoje dvije osnovne vrste elemenata.

Prva vrsta su **blok-elementi** *(block elements)*. Oni zauzimaju cijeli redak u kojem se nalaze. Moguće im je odrediti visinu i širinu. Čak i ako im se smanji širina, oni će i dalje zauzimati cijeli red tako da sljedeći element mora započeti u novome redu.



Primjeri takvih elemenata su `div`, `p` i `h1`. Element `div` služi za grupiranje sadržaja i drugih elemenata, element `p` služi za definiranje odlomka, dok element `h1` služi za postavljanje naslova (prve razine).

Element `div`, odlomak i naslov zauzet će cijeli redak, odnosno cjelokupnu širinu preglednika.

Druga vrsta su **linijski elementi** (*inline elements*). Ti su elementi prikazani unutar trenutačne linije teksta. Oni ne zauzimaju cijeli red nego onoliko prostora koliko zauzima njihov sadržaj.

Zanimljivosti i napomene

Ponašanje pojedinog elementa moguće je promijeniti korištenjem CSS-svojstva **display**.



Primjeri su takvih elemenata `span`, `em` i `strong`. Element `span` služi za označavanje dijela teksta (najčešće kako bi se na njega mogao primijeniti određeni stil), dok elementi `em` i `strong` također služe za označavanje, odnosno naglašavanje dijela teksta i prikazuju se s

predefiniranim stilom. Ovi elementi neće uzrokovati prebacivanje u novi red, već njihova veličina ovisi o sadržaju (prvenstveno tekstu) koji se unutar njih nalazi.

3.2. Naslovi – *h* elementi

Različite tematske cjeline dokumenta najčešće se odvajaju naslovima. HTML ima šest razina naslova i to `h1`, `h2`, `h3`, `h4`, `h5`, `h6`.

H1 je prva razina naslova i koristi se uglavnom za najvažnije naslove, naslove stranica, kao glavni naslov dokumenta i slično. Prikazuje se najvećom veličinom slova.

H6 je zadnja razina naslova i ispisuje se najmanjom veličinom slova.

Primjer:

```
<h1>Popis tečajeva Srca</h1>

<h2>Popis tečajeva u učionici</h2>

<h3>Tečajevi o osnovama uporabe računala i
Interneta</h3>

<h3>Tečajevi o web-tehnologijama, izradi web-stranica
i web-sjedišta</h3>

<h3>Drugi tečajevi</h3>

<h3>Radionice</h3>

<h2>Popis on-line tečajeva</h2>

<h3>Tečajevi o osnovama uporabe računala i
Interneta</h3>

<h3>Drugi tečajevi</h3>


<h3>Kratki tečajevi - prezentacije</h3>
```

Popis tečajeva Srca

Popis tečajeva u učionici

Tečajevi o osnovama uporabe računala i Interneta

Tečajevi o web-tehnologijama, izradi web-stranica i web-sjedišta

Drugi tečajevi

Radionice

Popis on-line tečajeva

Tečajevi o osnovama uporabe računala i Interneta

Drugi tečajevi

Kratki tečajevi - prezentacije

3.3. Odlomak – *p* element

Odlomci su tematske i logičke cjeline sadržaja koje u HTML-dokumentu definiraju blok elementom **odlomak** (engl. *paragraph*) `<p>`. Element `<p>` obavezno mora imati i završnu oznaku `</p>`.

Svaki preglednik prikazat će odlomak u novom redu, a između odlomka i ostalog sadržaja postavit će i razmak.

Primjer:

```
<p>Potrebno je hitno uvesti promjene! Zašto? Ako
ostane sve kao što je bilo do sada ne piše nam se
dobro u budućnosti.</p>

<p>Mislite danas na budućnost!</p>
```

Potrebno je hitno uvesti promjene! Zašto? Ako ostane sve kao što je bilo do sada ne piše nam se dobro u budućnosti.

Mislite danas na budućnost!

3.4. Prelazak u novi red – *br*

U slučaju da je unutar jednog odlomka potrebno započeti neki tekst u sljedećem redu (bez dodatnih razmaka između reda i ostalog teksta) koristit će se element za prelazak u novi red **br** (engl. *break row*).

Element
 nema završnu oznaku.

Primjer:

```
<p>Potrebno je hitno uvesti
promjene!<br>Zašto?<br>Ako ostane sve kao što je bilo
do sada ne piše nam se dobro u budućnosti.</p>

<p>Mislite danas na budućnost!</p>
```

Potrebno je hitno uvesti promjene!
Zašto?
Ako ostane sve kao što je bilo do sada ne piše nam se dobro u budućnosti.

Mislite danas na budućnost!

3.5. Vodoravna crta – *hr*

Kada je potrebno neki dio stranice logički odvojiti od prethodnog te to odvajanje dodatno naglasiti, može se koristiti **vodoravna crta – *hr***. Kao i prelazak u novi red, element vodoravna crta također nema završnu oznaku nego se koristi samo <hr>. Riječ je o blok elementu što znači da će se vodoravna crta uvijek postaviti u zaseban red.

Primjer:

```
<p>Potrebno je hitno uvesti
promjene!<br>Zašto?<br>Ako ostane sve kao što je bilo
do sada ne piše nam se dobro u budućnosti.</p>

<hr>

<p>Mislite danas na budućnost!</p>
```

Potrebno je hitno uvesti promjene!
Zašto?
Ako ostane sve kao što je bilo do sada ne piše nam se dobro u budućnosti.

Mislite danas na budućnost!

3.6. Razmak i drugi specijalni znakovi – * * i ostali HTML-entiteti

Postoje neki specijalni znakovi koji se koriste za oblikovanje i definiranje samog HTML-dokumenta, npr. šiljaste zagrade < i >, koje će HTML-dokument uvijek prepoznavati kao početak HTML-oznake. Dodatni primjer specijalnih znakova su i znakovi koji su inače sastavni dio HTML-dokumenta, kao što je npr. razmak. U slučaju kada se želi prikazati više od jednog razmaka za redom na *web*-stranici, potrebno je koristiti specijalne znakove, jer će HTML inače uvijek ignorirati sve osim jednog razmaka, tj. prikazat će se samo jedan razmak. Postavlja se pitanje kako te specijalne znakove zapisati kao dio teksta na *web*-stranici.

Takvi specijalni znakovi prikazuju se pomoću **HTML-entiteta** – posebnih kôdova za zapisivanje tih specijalnih znakova u HTML-dokument. Ti će se kôdovi na *webu* prikazati u onom obliku u kojem ih želimo prikazati.

Svaki od tih specijalnih znakova može se u HTML-dokumentu zapisati na dva načina; korištenjem alfanumeričkog kôda ili brojčanog kôda.

Specijalni HTML-znakovi

Kôd	Brojčani kôd	**Znak**	Opis
"	"	"	Dvostruki navodnici
&	&	&	engl. *Ampersand*
<	<	<	Manje od
>	>	>	Veće od
			Razmak (engl. *Non-breaking space*)
–	–	–	Mala crta
—	—	—	Velika crta
‘	‘	‘	Lijevi jednostruki navodnik
’	’	’	Desni jednostruki navodnik
‚	‚	‚	Zarez
“	“	"	Lijevi dvostruki navodnik
”	”	”	Desni dvostruki navodnik
„	„	"	Dvostruki zarez
‹	‹	‹	Lijevi uglati navodnik
›	›	›	Desni uglati navodnik
¢	¢	¢	Cent
£	£	£	Funta
¥	¥	¥	Yen
€	€	€	Euro

Zanimljivosti i napomene

Prilikom uporabe ovih znakova potrebno je uvijek provjeriti kako se prikazuju u *web*-pregledniku s obzirom na to da neki od fontova koji se koriste ne podržavaju ove znakove. U tom slučaju potrebno je u CSS-u postaviti onaj font koji će ove znakove moći prikazati.

Kôd	Brojčani kôd	**Znak**	Opis
¦	¦	¦	Prekinuta crta
§	§	§	Rubrika
©	©	©	engl. *Copyright*
®	®	®	engl. *Registered*
¼	¼	¼	Četvrtina
½	½	½	Polovina
¾	¾	¾	Tri četvrtine
™	™	™	engl. *Trade mark*

3.7. Predformatirani tekst – *pre*

U slučaju da se u nekom tekstu želi zadržati velik broj razmaka, zbog čega je korištenje specijalnih HTML-znakova nezgodno, ili se želi prikazati tekst koji ima točno onakvo oblikovanje kakvo je imao npr. u *Word* dokumentu iz kojeg je kopiran, može se koristiti **predformatirani tekst – *pre***.

Primjer:

```
<pre>
Prošla je berba, milo moje srce,
     I nema na trsju grozdova.
Na lozama se lišće trese kao ruke staraca.

Boje se u zemlju vraćaju,
     Boje se tope, i nestaju.
</pre>
```

```
Prošla je berba, milo moje srce,
    I nema na trsju grozdova.
Na lozama se lišće trese kao ruke staraca.

Boje se u zemlju vraćaju,
    Boje se tope, i nestaju.
```

3.8. Supskript i superskript – *sub* i *sup*

Za neke posebne znakove, kao što su kemijske i fizikalne formule ili za matematičke oznake indeksa i eksponenta potrebno je rabiti supskript ili superskript kako bi dio formule bio ispravno prikazan. Npr. ako na *web*-stranici treba napisati $E=MC^2$ koristit će se **superskript – sup**, dok će se za zapis H_20 koristi **supskript – sub**.

Primjer:

```
<p> Specijalna teorija relativnosti izražava ovaj
odnos rabeći formulu ekvivalentnosti mase i energije
E=MC<sup>2</sup></p>

<p>Količina CO<sub>2</sub> u atmosferi sve je
veća.</p>
```

> Specijalna teorija relativnosti izražava ovaj odnos rabeći formulu ekvivalentnosti mase i energije $E=MC^2$.
>
> Količina CO_2 u atmosferi sve je veća.

3.9. Stilski elementi – *b, i, u*

Ako se želi promijeniti grafički izgled nekog dijela dokumenta ili teksta koriste se **stilski HTML-elementi**.

Tako je tekst moguće podebljati, nakositi ili podcrtati korištenjem stilskih elemenata `<b>`, `<i>` i `<u>`.

Primjer:

```
<p>Ovo je <b>podebljani</b> tekst.</p>

<p>Ovo je <i>nakošeni</i> tekst.</p>

<p>Ovo je <u>podcrtani</u> tekst.</p>
```

> Ovo je **podebljani** tekst.
>
> Ovo je *nakošeni* tekst.
>
> Ovo je podcrtani tekst.

Zanimljivosti i napomene

Engleske riječi od kojih dolaze nazivi stilskih elemenata su:

- **Bold**<b>
- *Italic*<i>
- Underline<u>

3.10. Semantički elementi – strong, em, blockquote, q, cite, abbr, s

Postoje neki elementi čija namjena nije da promijene izgled ili strukturu *web*-stranica, nego da dijelovima stranice pridodaju dodatne informacije. Riječ je o **semantičkim HTML-elementima**. Oni služe za naglašavanje pojedinog dijela teksta, umetanje citata, označavanje kratica i slično.

Njih će *web*-preglednik često prikazivati na drugačiji način nego ostatak teksta, ali ih se zbog toga ne bi trebalo koristiti kao elemente za oblikovanje teksta.

Kada je to prikladno, preporučuje se koristiti semantičke elemente umjesto stilskih elemenata. Semantički elementi nose informaciju o značenju sadržaja te su zbog toga vrlo korisni za optimizaciju *web*-stranica. Osim ovoga, druga vrlo važna namjena semantičkih elemenata je olakšavanje korištenja *web*-stranica u programima koji nisu *web*-preglednici. Npr. čitači koje koriste slijepe i slabovidne osobe će semantičke elemente interpretirati na način da će npr. `<strong>` element snažno naglasiti.

Naglašavanje sadržaja

HTML-element **strong** (snažno) označava da sadržaj ima visoku važnost i da treba na njega obratiti posebnu pažnju. Element se prikazuje na isti način kao i stilski element `<b>` (*bold*), ali nosi dodatno značenje.

Primjer:

```
<p><strong>Pažnja:</strong> ne naginjite se kroz
prozor dok je vozilo u pokretu.</p>

<p>Ova igračka ima sitne odvojive dijelove i
<strong>nije prikladna za djecu mlađu od 5
godina.</strong></p>
```

> **Pažnja:** ne naginjite se kroz prozor dok je vozilo u pokretu.
>
> Ova igračka ima sitne odvojive dijelove i **nije prikladna za djecu mlađu od 5 godina.**

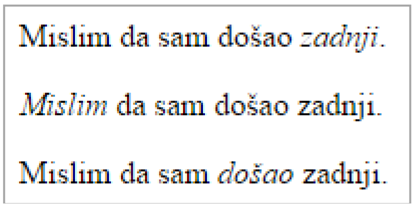
Element **em** (*emphasis*, naglašavanje) označava da bi naglasak trebao biti na tome dijelu teksta. Takav naglasak može suptilno mijenjati značenje rečenice. Preglednik će takav element prikazati na isti način kao i stilski element `<i>` (*italic*).

Primjer:

```
<p>Mislim da sam došao <em>zadnji</em>.</p>

<p><em>Mislim</em> da sam došao zadnji.</p>

<p>Mislim da sam <em>došao</em> zadnji.</p>
```



Citiranje i kratice (Definicije)

Blockquote se koristi za citiranje teksta. Riječ je o blok elementu koji se prikazuje kao uvučeni tekst. Za uvlačenje teksta koji nije citat, trebao bi se koristiti CSS.

Unutar `<blockquote>` elementa potrebno je koristiti element za oznaku paragrafa `<p>`.

```
<blockquote cite="http://citati.hr/citat-353">
<p>Čovjek je skitnica već po rođenju: ne zna otkud je
došao, ni kamo se zaputio.</p>
</blockquote>
```

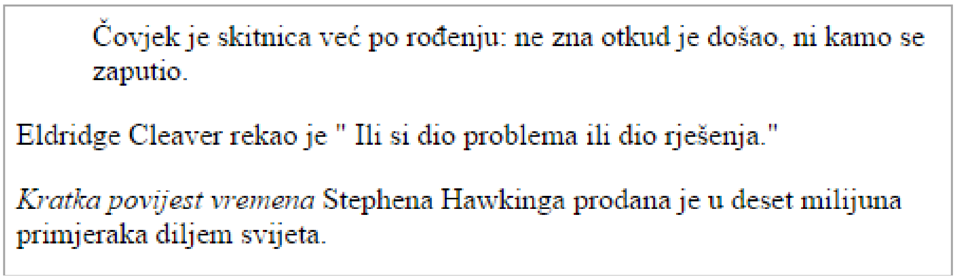
Za citiranje koje se nalazi usred rečenice potrebno je koristiti *inline* element **q**. Većina preglednika će oko teksta označenog s elementom `<q>` sama postaviti navodnike. No, neki neće pa je stoga potrebno provjeriti kako se citat prikazuje.

Oba elementa za citiranje; `<blockquote>` i `<q>` mogu imati atribut **cite.** Njegova je vrijednost URL koji vodi do izvora citata ili do stranice s više informacija o citatu.

```
<p>Eldridge Cleaver rekao je <q
cite="http://citati.hr/citat-2831">
Ili si dio problema ili dio rješenja.</q></p>
```

Ako se u tekstu referira na neko tuđe djelo kao što je knjiga ili istraživanje, moguće je označiti naziv tog djela HTML-elementom **cite** kako bi bilo jasno da je to izvor reference. *Web*-preglednik će tekst unutar oznaka `<cite>` prikazati kao nakošeni tekst.

```
<p><cite>Kratka povijest vremena</cite> Stephena
Hawkinga prodana je u deset milijuna primjeraka
diljem svijeta.</p>
```



Kada se koristi kratica ili akronim to je moguće naznačiti elementom **abbr**. U atribut **title** moguće je upisati puno značenje kratice ili akronima. Tekst u *web*-pregledniku izgleda nepromijenjeno.

Primjer:

```
<p>Posebno pozvano predavanje održat će <abbr
title="Professor">Prof.</abbr> Ray J. Paul pod
nazivom "Kakva vrsta sustava su zdravstveni
sustavi?"</p>
```

Posebno pozvano predavanje održat će Prof. Ray J. Paul pod nazivom „Kakva vrsta sustava su zdravstveni sustavi?“

Promjene sadržaja

Element `<s>` označava da nešto više nije točno ili relevantno, ali da i dalje ne bi trebalo biti obrisano s *web*-stranice. Na *web*-stranici taj se sadržaj prikazuje prekriženo. Najčešći bi primjer korištenja bio cijena proizvoda prije sniženja.

Primjer:

```
<p>Ženske zimske cipele</p>
<p><s>Cijena 1.500,00 kn</s></p>
<p>Sada samo 650,00 kn</p>
```

Ženske zimske cipele

~~Cijena 1.500,00 kn~~

Sada samo 650,00 kn

4. Popisi (liste)

Po završetku ovoga poglavlja polaznik će moći:

- *objasniti koja je razlika između numeriranog popisa i popisa s grafičkim oznakama*
- *objasniti čemu služi definicijska lista (popis)*
- *izraditi različite vrste lista (popisa) u HTML-u.*

Liste (popisi) su nizovi elemenata koji su međusobno povezani u smislenu grupu.Popisi po načinu označavanja i svrsi mogu biti:

- numerirani popisi – elementi su označeni rednim brojem
- popisi s grafičkim oznakama – elementi su označeni simbolom
- definicijski popisi – elementi popisa sastoje se od pojmova i njihovih definicija
- ugniježđeni popis – popisi koji imaju više razina.

U ovome poglavlju bit će prikazana izrada različitih tipova popisa u HTML-u.

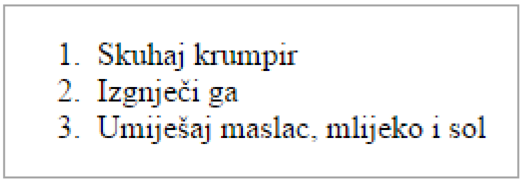
4.1. Numerirani popis – *ol*

Numerirani popisi su popisi čiji elementi su označeni rednim brojevima ili slovima. U praksi se najčešće koriste kada je potrebno navesti slijed elemenata u popisu, npr. za rang liste, upute za korištenje, kada je potrebno navesti slijed koraka i slično.

Numerirani popis izrađuje se uz pomoć HTML-elementa **`<ol>`** (engl. *Ordered list*), a za definiranje pojedinih elemenata liste tj. popisa, koristi se element **`<li>`** (engl. *list item*).

Primjer:

```
<ol>
      <li>Skuhaj krumpir</li>
      <li>Izgnječi ga</li>
      <li>Umiješaj maslac, mlijeko i sol</li>
</ol>
```



Atributi specifični za numerirani popis:

- `type` – stil numeriranog popisa; 1 (arapske brojke), a (mala slova abecede), A (velika slova abecede), i (male rimske brojke), I (velike rimske brojke). Ipak, preporučeno je da se za određivanje vrste numeracije koristi CSS.
- `start` – omogućuje promjenu početnog broja
- `reversed` – omogućuje numeraciju elemenata od većeg prema manjem broju.

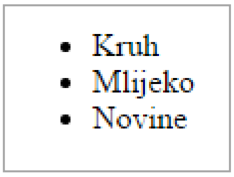
4.2. Popis s grafičkim oznakama – *ul*

Popisi s grafičkim oznakama su oni popisi čiji su elementi označeni simbolima ili znakovima. U praksi se najčešće koriste kada nije potrebno navesti slijed elemenata u popisu, tj. kada su svi elementi popisa jednako važni ili nemaju hijerarhijski odnos.

Popis s grafičkim oznakama izrađuje se pomoću HTML-elementa **<ul>** (engl. *unordered list*), a za definiranje pojedinih elemenata popisa koristi se element **<li>** (engl. *list item*).

Primjer:

```
<ul>
    <li>Kruh</li>
    <li>Mlijeko</li>
    <li>Novine</li>
</ul>
```



Prije se koristio atribut `type` da bi se odredila vrsta popisa s grafičkim oznakama, tj. da bi se odredilo hoće li se za oznaku sljedećeg elementa koristiti crtica, kružić, strelica ili neki drugi simbol, no sada je preporučeno da se za određivanje vrste simbola koristi CSS.

4.3. Definicijski popis – dl

Definicijski popisi su oni popisi u kojima se svaki element popisa sastoji od pojma i definicije koji najčešće dolaze u parovima, tj. **definicijskog izraza** `<dt>` (engl. *Definition term*) i same **definicije** `<dd>` (engl. *Definition description*). Ponekad je ipak moguće da jedan pojam ima više definicija, ili da se više definicija veže uz jedan pojam. Definicijski se popisi mogu koristiti za kratke, specijalizirane popise i objašnjenja te rječnike na *web*-stranicama.

Definicijski popis izrađuje se pomoću HTML-elementa <dl> (engl. *Definition list*). Popis će se prikazati tako da će definicija uvijek biti malo uvučenija u odnosu na definicijski izraz, a elementi definicijskog popisa neće biti dodatno označeni ni brojevima ni simbolima.

Primjer:

```
<dl>
      <dt>Pobrojana lista </dd>
      <dd>Elementi liste označeni su rednim
      brojem</dd>
      <dt>Nepobrojana lista</dt>
      <dd>Elementi liste označeni su simbolom</dd>
      <dt>definicijska lista</dt>
      <dd>Elementi liste sastoje se od pojmova i
      njihovih definicija<dt>
      <dt>Ugniježđene liste</dt>
      <dd>Liste koje imaju više razina.</dd>
</dl>
```

Pobrojana lista
    Elementi liste označeni su rednim brojem
Nepobrojana lista
    Elementi liste označeni su simbolom
definicijska lista
    Elementi liste sastoje se od pojmova i njihovih definicija
Ugniježđene liste
    Liste koje imaju više razina.

4.4. Ugnježđivanje popisa

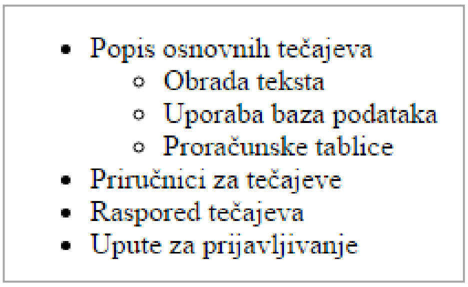
Unutar <li> elementa moguće je dodati novi popis ako se želi izraditi ugniježđeni popis. Ugniježđeni se popisi koriste kako bi se dodatno raspisao neki element popisa. Vrlo se često koristi i kao način za izradu navigacije za *web*-stranicu.

Ugniježđeni će se popis u pregledniku prikazati uvučenije od osnovnog popisa.

Primjer:

```
<ul>
      <li>Popis osnovnih tečajeva</li>
      <ul>
            <li>Obrada teksta</li>
<li>Uporaba baza podataka</li>
<li>Proračunske tablice</li>
```

```
</ul>
<li>Priručnici za tečajeve</li>
<li>Raspored tečajeva</li>
<li>Upute za prijavljivanje</li>
</ul>
```



Popisi se mogu ugnježđivati do proizvoljne razine.

4.5. Vježba: Oblikovanje teksta i popisi

1. Izradite novi HTML-dokument i spremite ga u mapu **C201/vjezbe/vjezba1** pod nazivom **kontakt.html**.

2. U ovom dokumentu napišite `DOCTYPE` deklaraciju i osnovnu strukturu HTML-dokumenta (elemente `<html>`, `<head>` i `<body>`).

3. Otvorite datoteku **kontakt.txt** koja se nalazi u mapi **C201/vjezbe/vjezba1** i kopirajte njezin sadržaj unutar elementa `<body>`.

4. Otvorite HTML-dokument iz koraka 1. u *web*-pregledniku i provjerite prikazuju li se hrvatski znakovi ispravno.

5. Unutar elementa `<head>` dodajte element `<meta>` i pomoću njega postavite ispravnu kôdnu stranicu.

6. Unutar elementa `head` dodajte element `<title>` i pomoću njega postavite naslov stranice u pregledniku (npr. "Tečajevi Srca").

7. Oblikujte dokument **kontakt.html** tako da izgleda kao odgovarajući dokument iz mape **rjesenja**(**C201/rjesenja/vjezba1/kontakt.html).**

Dodatni dio vježbe:

8. Izradite novi HTML-dokument **index.html** na temelju datoteke **index.txt** koji se nalazi u mapi Rjesenja (ponavljajući korake iz zadataka 1. – 7.).

9. Izradite novi HTML-dokument **popis.html** na temelju datoteke **popis.txt** koji se nalazi u mapi Rjesenja (ponavljajući korake iz zadataka 1. – 7.). Za nastavljanje brojanja u drugom numeričkom popisu upotrijebite atribut `start`.

   U vježbi se koriste HTML-elementi `<h1>`, `<h2>`,`<h3>`,`<p>`, `<br>`, `<hr>`, `<strong>`, `<em>`, `<ul>`, `<ol>`, `<li>`,`<sup>` te HTML entitet `©`.

5. Linkovi

Po završetku ovog poglavlja polaznik će moći:

- *objasniti što su linkovi i čemu služe*
- *izraditi više vrsta linkova u HTML-dokumentu (linkovi koji se otvaraju u novom prozoru, linkovi koji vode s jedne stranice na drugu, linkovi koji vode s jedne stranice na neko drugo mjesto na istoj toj stranici…)*
- *objasniti kako linkovi utječu na funkcionalnost web-sjedišta.*

Osim samoga prikazivanja sadržaja, glavna odlika svake *web*-stranice je mogućnost povezivanja dijelova sadržaja te prelazak s jedne stranice na drugu. Bez ovoga svojstva svake *web*-stranice, mogućnost pregledavanja stranica ili *surfanje* ne bi uopće postojalo. Način na koji se s jedne stranice može preći na drugu odvija se pomoću **linka** ili veze (poveznice).

Postoji nekoliko vrsta linkova:

- linkovi koji vode s jedne *web*-stranice na drugu
- linkovi koji vode s jedne stranice na drugu, unutar istog *web*-sjedišta
- linkovi koji vode s jednog dijela teksta na drugi dio teksta na istoj stranici
- linkovi koji se otvaraju u novom prozoru preglednika
- linkovi koji pokreću program za slanje e-pošte.

5.1. Link i njegovi atributi – *a* element

Linkovi se izrađuju korištenjem HTML-elementa `<a>`. Bilo kakav tekst (ili neki drugi element kao npr. slika) koji se nalazi između početne oznake `<a>` i završne oznake `</a>` postat će link, što znači da će se na taj tekst ili sliku moći kliknuti mišem.

Klikom na link prelazi se na neko drugo mjesto. O kojem je mjestu riječ, definira se korištenjem atributa `href`. Atribut mora imati formu oblika `href="URL"`.

Primjer:

```
<a href="http://www.srce.hr">Sveučilišni računski
centar Srce</a>
```

Tekst između početne i završne oznake je tekst linka i on će se na *web*-stranici prikazati tako kako je napisan. Gdje god je moguće tekst linka bi trebao biti jasan i objašnjavati na koju stranicu link vodi, a trebalo bi se izbjegavati korištenje riječi kao što su "više" ili "ovdje".

Zanimljivosti i napomene

Zbog prilagodbe *weba* za osobe s invaliditetom, potrebno je linkove učiniti razumljivima i izvan konteksta, same za sebe. Pri tome nije potrebno tekstom dodatno naglašavati da je riječ o linku.

Kao tekst linka nikada se ne bi trebao navoditi cijeli URL, zbog toga što čitači koje koriste slijepe i slabovidne osobe, linkove čitaju slovkajući pa bi u ovom slučaju čitali „duplo v, duplo v, duplo v…“.

Zanimljivosti i napomene

Uniform Resource Locator (URL) je putanja do nekog sadržaja na *webu* te se još naziva i *web*-adresa. Taj sadržaj može biti tekst, slika, HTML-dokument, neka datoteka…

Svaka stranica, dokument i slika na *webu* imaju svoj jedinstveni URL.

Mnogo ljudi pregledava veće dijelove teksta na pojedinoj stranici tražeći samo linkove, pa je iz tog razloga važno da su linkovi jasno označeni i vidljivi. Prilikom osmišljavanja teksta za link potrebno je pokušati razmišljati iz uloge osobe koja gleda *web*-stranicu i smisliti koje bi riječi netko vezao uz stranicu na koju link ide. Npr., ako je riječ o linku na neki turistički objekt ili smještaj u Splitu, bolje bi bilo da tekst linka bude "hotel u Splitu", umjesto "gdje odsjesti" i slično.

Ako link vodi na neko tuđe *web*-sjedište, vrijednost atributa href treba biti **apsolutni URL**. Apsolutni URL je ona *web*-adresa koja bi se upisala u preglednik kada se želi otvoriti neka stranica. Prvi dio takvog URL-a je ime domene određenog *web*-sjedišta, a može još imati i putanju do točno određene stranice.

Primjer:

```
<a href="http://www.srce.hr/kontakt.html">Kontakt</a>
```

Ako link vodi na neku stranicu koja je dio našeg *web*-sjedišta onda će vrijednost atributa href biti **relativni URL**. Relativni URL nema ime domene, nego samo ime stranice na koju vodi i po potrebi putanju do te stranice. Npr., ako su sve stranice nekog *web*-sjedišta u istoj mapi, onda će relativni URL biti samo ime dokumenta na koji se želi linkati.

Primjer:

```
<a href="kontakt.html">Kontakt</a>
```

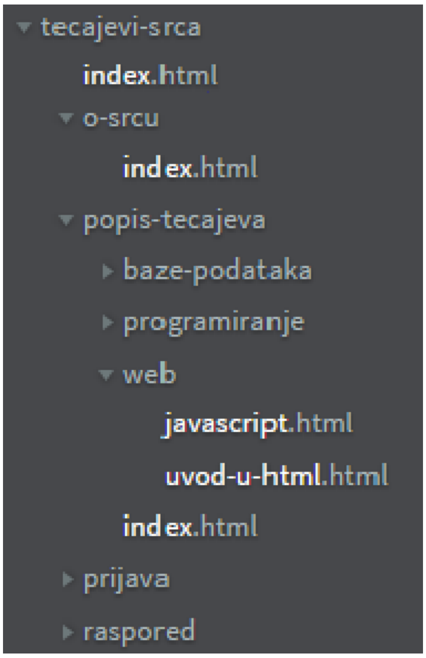
5.2. Putanje i struktura *web*-sjedišta

Zanimljivosti i napomene

U slučaju da se *web*-stranice izrađuju pomoću CMS sustava, postoji mogućnost da svaka stranica nije pojedinačni HTML-dokument.

Umjesto toga, ovi sustavi često koriste po jedan predložak za jednu vrstu stranice (vijesti, proizvodi, blog…). U slučaju da se rade izmjene koje se ne tiču samoga sadržaja na nekome od predložaka, postoji mogućnost da će se izmjeniti i sam predložak pa samim time i sve druge stranice koje koriste taj predložak.

Radi veće preglednosti i lakše organizacije datoteke *web*-sjedišta najčešće su organizirane u hijerarhijsku strukturu. Ta se struktura postiže na način da se različiti segmenti *weba* spremaju u vlastite mape.



Ova je struktura razgranata kao stablo i proteže se na nekoliko razina, te je zbog toga pisanje relativnih URL-ova kompleksnije.

U primjeru koji je prikazan s desne strane, na prvoj razini nalazi se početna stranica *index.html.*

Na drugoj razini nalazi se datoteka *index.html* u mapi *o-srcu* te datoteka *index.html* u mapi *popis-tecajeva*. Na trećoj razini nalaze se datoteke *javascript.html* i druge, koje se nalaze unutar mape *popis-tecajeva/web.*

Početna stranica *web*-sjedišta najčešće se naziva *index.html*. Ako je imenovana na taj način, početna će stranica biti dostupna putem URL-a *www.domena.hr* (nije potrebno pisati punu putanju *www.domena.hr/index.html*).

Slično, ako se unutar mape *o-srcu* nalazi datoteka *index.html*, to će biti početna datoteka za tu mapu. Ona će biti dostupna putem URL-a *www.domena.hr/o-srcu* (nije potrebno pisati punu putanju *www.domena.hr/o-srcu/index.html*).

Korištenjem relativnih URL-ova prilikom izrade *weba* omogućava se linkanje sa stranice na stranicu koje je funkcionalno i na lokalnom računalu, bez imena domene. Na ovaj se način može napraviti navigacija za cijeli *web*. Ta će navigacija raditi dok su svi HTML dokumenti na lokalnome računalu, ali i kad se prebace na server, pod uvjetom da se ne mijenja struktura mapa i lokacija dokumenata u mapama.

Ako su sve stranice od kojih se *web* sastoji u jednoj mapi, onda je dovoljno samo navesti ime stranice u *href* atributu *a* elementa. Međutim, ako su stranice organizirane po mapama, potrebno je odgovarajućom putanjom naznačiti pregledniku kako će doći od stranice na kojoj jest, do stranice na koju treba ići.

Putanja	Primjer
Putanja do dokumenta u istoj mapi	Link sa stranice Osnove Javascripta (popis-tecajeva/web/javascript.html) na stranicu Uvod u HTML (popis-tecajeva/web/uvod-u-html.html) Uvod u HTML
Putanja do dokumenta u mapi ispod	Link sa stranice Popis tečajeva (*popis-tecajeva/index.html)* na stranicu Uvod u HTML (*popis-tecajeva/web/uvod-u-html.html)* Uvod u HTML
Putanja do dokumenta dvije mape ispod	Link s početne stranice (*index.html*) na stranicu Uvod u HTML (*popis-tecajeva/web/uvod-u-html.html)* Uvod u HTML
Putanja do dokumenta u mapu iznad	Link sa stranice Uvod u HTML (*popis-tecajeva/web/uvod-u-html.html)* na stranicu Popis tečajeva (*popis-tecajeva/index.html)* Tečajevi
Putanja do dokumenta dvije mape iznad	Link sa stranice Uvod u HTML (*popis-tecajeva/web/uvod-u-html.html)* na početnu stranicu (*index.html*) Tečajevi

Putanja	Primjer
Putanja do dokumenta na drugom dijelu stabla	Link sa stranice O srcu (*o-srcu/index.html)* na stranicu Uvod u HTML (*popis-tecajeva/web/uvod-u-html.html)* `<a href="../popis-tecajeva/web/uvod-u-html.html">Uvod u HTML</a>`

5.3. Otvaranje linka u novom prozoru

Svi linkovi se zadano otvaraju u istome prozoru u kojem se nalaze. Ako se želi omogućiti da se link otvara u novom prozoru potrebno je koristiti `target` atribut HTML-elementa `a` i postaviti mu vrijednost na `_blank`.

Primjer:

Sveučilišni računski centar Srce

Najčešći razlog za ovakvo otvaranje linka je u slučaju da link vodi s naših *web*-stranica na neke druge stranice. U tom će slučaju posjetitelj nakon što pregleda novi sadržaj i zatvori prozor, "izgubiti" prvobitnu stranicu. Ako se pak link otvori u drugom prozoru, posjetitelj će nakon zatvaranja prozora biti vraćen na polaznu *web*-stranicu.

5.4. *Mailto* i *tel* linkovi

Posebna vrsta linkova su linkovi koji pokreću vanjske aplikacije za slanje e-pošte ili za telefoniranje. Riječ je o **mailto** i **tel** linkovima.

Zanimljivosti i napomene

Protokoli mailto, tel i sms nisu dio http protokola ali su vrlo popularni zbog svoje korisnosti.

Neki mobilni preglednici automatski telefonske brojeve pretvaraju u linkove kojima se može pokrenuti razgovor, ali korištenjem navedenih linkova se ovakva mogućnost osigurava.

Da bi se izradio link koji pokreće korisnički program za slanje e-pošte i popunjava polje s adresom, koristi se HTML-element `a`. Ovoga puta atribut `href` započinje s `mailto`:, a odmah zatim slijedi adresa e-pošte na koju se želi poslati e-pošta.

Primjer:

```
Za više pitanja obratite se <a
href="mailto:helpdesk@srce.hr">Helpdesku Srca</a>.
```

U novije vrijeme postalo je uobičajeno pregledavati Internet na mobitelu, pa se, kako bi se takvim korisnicima olakšalo telefoniranje, koristi protokol `tel`, koji omogućuje automatsko pokretanje telefoniranja s već unesenim telefonskim brojem.

Primjer:

```
Ako imate pitanja nazovite nas na <a
href="tel:+38516165165">6165-165</a>.
```

Na isti se način može koristiti i protokol `sms`.

5.5. Link na određeni dio stranice – sidro

Na vrhu vrlo duge stranice s puno sadržaja može se dodati pregled sadržaja na stranici. U tom će slučaju svaki element u sadržaju biti link koji vodi na točno određeni dio stranice, na koji se odnosi. Također, kod dugih stranica, nakon pojedinog poglavlja može se dodati i link koji korisnika vraća na vrh stranice, tj. na pregled sadržaja.

Prije nego što je moguće napraviti link na određeni dio stranice, prvo se treba identificirati mjesto (ili sidro) na koje će link voditi. Ovo se radi na način da se elementu na koji želimo da link vodi, doda atribut `id`. `Id` je atribut koji se može dodati svakom HTML-elementu, proizvoljan je, ali bi trebao započinjati slovom ili podcrtom (_). Nijedan `id` atribut ne smije imati istu vrijednost.

Jednom kada je određena točka na koju će link voditi, može se izraditi i sam link. Da bi link išao na element koji koristi `id` atribut koristi se `<a>` element, ali ovaj se put u atribut `href` doda simbol ljestva (#) i vrijednost `id` atributa na koji se želi stvoriti veza.

Primjer:

```
<p id="vrh">Popis tečajeva:</p>
<ul>
     <li><a href="#sql">Uvod u SQL</a></li>
     <li><a href="#ruby">Uvod u programski jezik
     Ruby</a></li>
<li><a href="#PHP">Uvod u PHP i MySQL</a></li>
</ul>

<h2 id="sql">Uvod u SQL</h2>
<p>D300</p>
<p>Objekti, relacije i baze podataka; kompleksne
selekcije; operacije nad relacijama; elementi teorije
normalizacije; oblikovanje relacijske baze
podataka.</p>

<h2 id="ruby">Uvod u programski jezik Ruby</h2>
<p>D400</p>
<p>Osnove jezika Ruby, izrazi, klase i objekti, rad s
datotekama, upravljanje greškama, moduli, testiranje
programskog koda.</p>

<h2 id="PHP">Uvod u PHP i MySQL</h2>
<p>C200</p>
<p>Model klijent-poslužitelj, varijable, operatori,
uvjetne strukture, polja, petlje, funkcije, ugrađene
funkcije PHP-a, obrasci, prijenos podataka između
```

```
skripti, rad s datotekama, slanje poruke elektroničke
pošte, rad s bazom podataka MySQL, sjednice,
autentikacija korisnika.</p>

<p>Povratak na <a href="#vrh">vrh</a>.</p>
```

Ako se želi napraviti link prema točno određenome mjestu na nekoj drugoj stranici i to je moguće korištenjem ove metode. Važno je da mjesto na koje se želi povezati ima definiran `id` atribut. U primjeru će biti prikazano kako se linka s neke druge stranice na opis tečaja Uvod u PHP i MySQL.

Primjer:

```
<p>Detalje o tečaju Uvod u PHP i MySQL moguće je
pronaći na stranici s<a href="www.domena.hr/popis-
tecajeva#PHP">popisom tečajeva</a>.</p>
```

U slučaju da se želi napraviti link na točno određeno mjesto neke stranice na nekome drugom *web*-sjedištu, i to je moguće ako je dostupan `id` atribut te lokacije.

U ova se dva slučaja može koristiti relativni URL za link na drugu stranicu našega *web*-sjedišta ili apsolutni URL za link na tuđe *web*-sjedište.

6. Slike

Po završetku ovog poglavlja polaznik će moći:

- *odabrati i objaviti sliku na web-stranici*
- *odabrati format slike koji je najbolji za objavu na web-stranici*
- *postaviti opis i druga svojstva slike.*

Slike i ostali grafički elementi vrlo su važan dio sadržaja svake *web*-stranice, bilo da nose informaciju ili služe samo u estetske svrhe.

Određene vrste slika mogu vrlo brzo ostaviti određeni dojam o *web*-stranici i prije nego što se stigne pročitati sadržaj stranice. U slučaju nemogućnosti izrade autorskih fotografija i grafika, iste je moguće kupiti preko različitih stranica koje prodaju slike, tzv. *stock images*.

Zanimljivosti i napomene

Prilikom pripreme slika za objavu na *web*-stranicama potrebno je misliti na „težinu" slika, tj. optimizirati fizičku veličinu slike tako da ona bude što manja, bez da se naruši njezina kvaliteta.

Slike na *web*-stranicama mogu ispunjavati različite svrhe pa je prilikom odabira slika za *web*-stranicu potrebno obratiti pažnju na to da:

- budu relevantne
- prenose određenu informaciju
- dočaravaju odgovarajuću atmosferu
- budu jasne
- budu u skladu s ostalim bojama na *web*-stranici.

6.1. Slika – *img* element

Sve slike koje se koriste na *web*-stranici trebaju prethodno biti spremljene na nekoj lokaciji unutar *web*a. Uobičajena je praksa da se za spremanje slika otvori jedna mapa, najčešće imena "Slike", te u njoj odgovarajuće podmape radi lakšega snalaženja.

Za umetanje slika bilo kojega formata u HTML-kôdu rabi se element `<img>`. Element nema završnu oznaku.

Obavezni atribut elementa `img` je `src` koji govori pregledniku putanju do slike. Obično se koristi relativni URL, s obzirom na to da se slika nalazi unutar mape na serveru.

Primjer:

```
<img src="slike/osnovni_tecajevi.png" />
```

Atribut `alt` daje opis slike u slučaju da se slika iz bilo kojeg razloga nije u stanju učitati u pregledniku. Važno je da opis bude točan i jasan, jer je riječ o tekstu koji će se, kada slijepe i slabovidne osobe pristupaju *web*-stranici, pročitati pomoću specijaliziranih čitača.

Primjer:

```
<img src="slike/osnovni_tecajevi.png" alt="Polaznici
tečajeva"/>
```

U slučaju da slika nema nikakvo značenje, već služi samo kao dizajnerski element (npr. neka slika određene boje koja se koristi kao razdjelnik sadržaja), svejedno bi trebala imati `alt` atribut, bez opisa.

Primjer:

```
<img src="slike/crta.png" alt="" />
```

Atribut `title` omogućuje prikazivanje dodatnih informacija o slici kada se mišem pređe preko slike.

Primjer:

```
<img src="slike/osnovni_tecajevi.png" alt="Polaznici
tečajeva" title="Polaznici tečajeva u prostorijama
Sveučilišnog računskog centra"/>
```

Atributi koji služe za određivanje veličine slike u pikselima su `height` i `width`. Ovi se atributi mogu koristiti, ali preporuča se prethodna optimizacija slika, tj. spremanje slika točne veličine te određivanje veličine slika pomoću CSS-a. Također, nepažljivim korištenjem oba atributa istovremeno, može doći do izobličenja slike.

Primjer:

```
<img src="slike/osnovni_tecajevi.png" alt="Polaznici
tečajeva" title="Polaznici tečajeva u prostorijama
Sveučilišnog računskog centra" width="120"
height="150"/>
```

Smještanje slika uz tekst

Stavljanje <img> elementa u kodu utječe na to gdje će slika biti smještena na stranici, tj. u odnosu na tekst.

Slika može biti smještena:

- prije odlomka
- na samome početku odlomka
- u sredini odlomka.

Ako se slika smješta prije odlomka, koji je blok element, slika će ostati u vlastitome retku, a odlomak će započeti u sljedećem retku jer blok element zauzima cijelu širinu stranice.

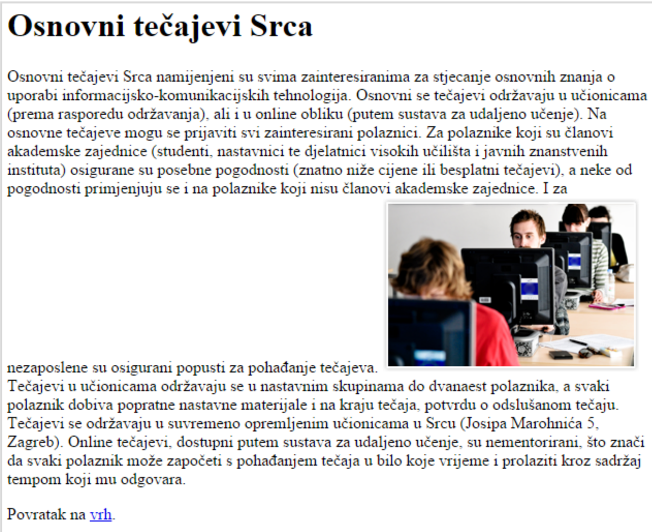


Osnovni tečajevi Srca

Osnovni tečajevi Srca namijenjeni su svima zainteresiranima za stjecanje osnovnih znanja o uporabi informacijsko-komunikacijskih tehnologija. Osnovni se tečajevi održavaju u učionicama (prema rasporedu održavanja), ali i u online obliku (putem sustava za udaljeno učenje). Na osnovne tečajeve mogu se prijaviti svi zainteresirani polaznici. Za polaznike koji su članovi akademske zajednice (studenti, nastavnici te djelatnici visokih učilišta i javnih znanstvenih instituta) osigurane su posebne pogodnosti (znatno niže cijene ili besplatni tečajevi), a neke od pogodnosti primjenjuju se i na polaznike koji nisu članovi akademske zajednice. I za nezaposlene su osigurani popusti za pohađanje tečajeva. Tečajevi u učionicama održavaju se u nastavnim skupinama do dvanaest polaznika, a svaki polaznik dobiva popratne nastavne materijale i na kraju tečaja, potvrdu o odslušanom tečaju. Tečajevi se održavaju u suvremeno opremljenim učionicama u Srcu (Josipa Marohnića 5, Zagreb). Online tečajevi, dostupni putem sustava za udaljeno učenje, su nementorirani, što znači da svaki polaznik može započeti s pohađanjem tečaja u bilo koje vrijeme i prolaziti kroz sadržaj tempom koji mu odgovara.

Povratak na vrh.

No, ako je slika smještena unutar odlomka, bilo na samome početku ili u sredini, tekst u odlomku ići će oko slike.



Osnovni tečajevi Srca

Osnovni tečajevi Srca namijenjeni su svima zainteresiranima za stjecanje osnovnih znanja o uporabi informacijsko-komunikacijskih tehnologija. Osnovni se tečajevi održavaju u učionicama (prema rasporedu održavanja), ali i u online obliku (putem sustava za udaljeno učenje). Na osnovne tečajeve mogu se prijaviti svi zainteresirani polaznici. Za polaznike koji su članovi akademske zajednice (studenti, nastavnici te djelatnici visokih učilišta i javnih znanstvenih instituta) osigurane su posebne pogodnosti (znatno niže cijene ili besplatni tečajevi), a neke od pogodnosti primjenjuju se i na polaznike koji nisu članovi akademske zajednice. I za nezaposlene su osigurani popusti za pohađanje tečajeva. Tečajevi u učionicama održavaju se u nastavnim skupinama do dvanaest polaznika, a svaki polaznik dobiva popratne nastavne materijale i na kraju tečaja, potvrdu o odslušanom tečaju. Tečajevi se održavaju u suvremeno opremljenim učionicama u Srcu (Josipa Marohnića 5, Zagreb). Online tečajevi, dostupni putem sustava za udaljeno učenje, su nementorirani, što znači da svaki polaznik može započeti s pohađanjem tečaja u bilo koje vrijeme i prolaziti kroz sadržaj tempom koji mu odgovara.

Povratak na vrh.

Slika je *inline* element što znači da će zauzeti onoliko mjesta na stranici koliko joj je potrebno, a zatim će se nastaviti prikazivati sljedeći element.

Više o blok i *inline* elementima bilo je riječi u poglavlju 3.1. Blok i *inline* elementi.

Zanimljivosti i napomene

Odmicanje slova od slike može se napraviti korištenjem CSS-svojstava `padding` i `margin`.

Ako se želi cijeli tekst omotati oko slike onda se koristi CSS-svojstvo `float`.

Smještanje slika unutar odlomka po horizontali ili vertikali izvodi se korištenjem CSS-a, ali zbog toga što se često može pronaći na starijim stranicama, u tablici je dan pregled smještanja slike uz pomoć atributa `align`.

Vrijednost atributa align	HTML-kôd	Primjer
Left	`<img src="slike/osnovni_tecajevi.png" align="left"/>`	
Right	`<img src="slike/osnovni_tecajevi.png" align="right"/>`	
Top	`<img src="slike/osnovni_tecajevi.png" align="top"/>`	
Middle	`<img src="slike/osnovni_tecajevi.png" align="middle"/>`	
Bottom	`<img src="slike/osnovni_tecajevi.png" align="bottom"/>`	

6.2. Formati slike za *web*

Slike namijenjene za objavu na *web*-stranicama trebale bi uvijek biti snimljene u onoj veličini u kojoj će se pojavljivati na samoj *web*-stranici, prije nego li ih se pohrani na poslužiteljskome računalu. Ako se slika sprema u manjoj veličini od one koja će biti prikazana, postoji mogućnost da će prikazom u planiranoj veličini na *webu* izgubiti na kvaliteti ili će biti izobličena. Ako je slika veća od veličine predviđene za *web*, sporije će se učitavati i korisniku će povećati količinu potrošenoga internetskog prometa.

U slučaju da na postojećemu *webu* treba promijeniti sliku, potrebno je prvo provjeriti kolika je bila prethodna slika. To se može napraviti klikom desne tipke miša na sliku i otvaranjem u novome prozoru te provjerom informacija o slici ili na način da se provjeri veličina spremljene slike na serveru.

Zanimljivosti i napomene

Neki sustavi za upravljanje sadržajem imaju dodatne mogućnosti i alate za pregledavanje veličine postojeće slike na *web*-stranici.

Osim ispravne veličine, važno je da slika bude spremljena i u ispravnome grafičkom formatu. *Web*-stranice uglavnom koriste formate *JPEG*, *GIF, SVG* i *PNG*. Odabir formata je važan jer odabirom krivoga formata može doći do gubitka oštrine ili nepotrebnoga opterećenja diska, korisničkoga prometa te sporoga učitavanja slike.

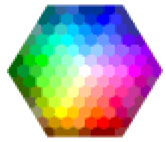
- **JPEG/JPG** je format koji se koristi uglavnom za prikaz fotografija, tj. slika koje u sebi imaju veliku količinu boja. JPEG se koristi i za crno-bijele, tj. dvobojne slike koje imaju puno prijelaza.

  Ovaj format može vjerno prikazati sliku u milijunima boja, zbog toga što koristi mogućnost sažimanja s gubicima. Sažimanje se još naziva i optimizacija slike. U slučaju da se slika previše sažme, izgubit će na kvaliteti i postoji mogućnost da ne izgleda dobro. Sažimanje slike i smanjivanje njezinih dimenzija u pikselima nisu povezani, iako će se smanjivanjem slika uvijek zauzimati i manje prostora.

- **GIF** je format koji se koristi za prikaz slika koje imaju veće plohe iste boje kao što su ilustracije, logotipi i slično. Te su plohe uglavnom jasno razdvojene od ploha drugih boja. GIF prikazuje do 256 boja i ne zahtijeva puno prostora na disku.

- Animirani GIF je format koji prikazuje nekoliko sličica za redom te na taj način stvara animaciju.



- **SVG** je noviji grafički format koji omogućuje objavljivanje slika u vektorskome formatu. Vektorski format omogućuje povećavanje grafika bez ikakvog gubitka kvalitete. Ovo je noviji format pa se može dogoditi da ga pojedini preglednik ne podržava.



- **PNG** je format koji podržava transparentnost, bilo na dijelu slike (potpuna prozirnost nekih dijelova) ili na cijeloj slici. Kao i GIF, zahtijeva vrlo malo prostora na disku. Često se koristi za dugmad i jednostavne slike, a s obzirom na to da omogućuje i gradaciju prozirnosti vrlo je popularan format za prikaz sjena nekih elemenata na *webu*.



6.3. Opis slike – *figure* i *figcaption* elementi

Slike na *web*-stranicama mogu imati i opis. Kako bi opis i slika uvijek bili zajedno, uveden je element <figure>. Unutar elementa <figure > moguće je imati i više od jedne slike, pod uvjetom da sve slike imaju isti opis.

Sam opis slike piše se unutar elementa <figcaption>.

Preglednici ponekad sadržaj unutar elementa figure prikazuju uvučeno.

Primjer:

```
<figure>
<img src="slike/osnovni_tecajevi.png" alt="Polaznici
tečajeva" title="Polaznici tečajeva u prostorijama
Sveučilišnog računskog centra"/>
<br>
<figcaption>Polaznici na tečaju u Srcu</figcaption>
</figure>
```

Osnovni tečajevi Srca



Polaznici na tečaju u Srcu

Osnovni tečajevi Srca namijenjeni su svima zainteresiranima za stjecanje osnovnih znanja o uporabi informacijsko-komunikacijskih tehnologija. Osnovni se tečajevi održavaju u učionicama (prema rasporedu održavanja), ali i u online obliku (putem sustava za udaljeno učenje). Na osnovne tečajeve mogu se prijaviti svi zainteresirani polaznici. Za polaznike koji su članovi akademske zajednice (studenti, nastavnici te djelatnici visokih učilišta i javnih znanstvenih instituta) osigurane su posebne pogodnosti (znatno niže cijene ili besplatni tečajevi), a neke od pogodnosti primjenjuju se i na polaznike koji nisu članovi akademske zajednice. I za

6.4. Vježba: Linkovi i slike

1. Otvorite postojeći HTML-dokument **index.html** u mapi **C201/vjezbe/vjezba2**.
2. Tekst "Sveučilišni računski centar" u prvom odlomku pretvorite u link koji vodi na adresu http://www.srce.unizg.hr. Neka se link otvara u novom prozoru.
3. Tekst "Raspored tečajeva" u pretposljednjem odlomku pretvorite u link koji vodi na stranicu **raspored.html** koja se nalazi u istoj mapi (**C201/vjezbe/vjezba2**). Link se treba otvarati u istom prozoru.
4. E-mail adresu *edu@srce.hr* u posljednoj rečenici pretvorite u link koji otvara program za slanje e-maila (npr. *Microsoft Outlook*) i započinje novu e-mail poruku na tu adresu.
5. Otvorite postojeći HTML-dokument **kontakt.html** u mapi **C201/vjezbe/vjezba2**.
6. Telefonski broj pretvorite u link koji će na mobilnim uređajima napraviti poziv na taj broj (navođenjem protokola *tel*).
7. E-mail adresu *edu@srce.hr* pretvorite u link koji otvara program za slanje e-maila.
8. Pri vrhu stranice (poslije elementa `h1`) dodajte popis s grafičkim oznakama koji će sadržavati četiri stavke. Ovaj popis sadržavat će navigaciju između stranica, pa unutar elemenata `li` postavite linkove "O tečajevima", "Popis tečajeva", "Raspored tečajeva" i "Kontakt" koji će voditi na stranice **index.html**, **popis.html, raspored.html** i **kontakt.html** koji se nalaze u istoj mapi (**C201/vjezbe/vjezba2**).

9. Isti popis s grafičkim oznakama koji sadrži navigacijske linkove kopirajte i postavite na ostalim stranicama u mapi (**index.html**, **popis.html, raspored.html**). Provjerite da navigacija između stranica radi.
10. Naslov `h1` zamijenite s elementom `img` kojim ćete prikazati sliku **logo.png** iz mape **C201/vjezbe/vjezba2/slike** na svim stranicama u mapi (**index.html**, **popis.html, raspored.html**). Ova će slika predstavljati logo stranice, pa joj postavite i tekst "Tečajevi Srca" u atribut `alt`.
11. Oko slike postavite element `a` kako bi slika postala link na početnu stranicu (**index.html**).
12. Ovu sliku i `a` element koji je obuhvaća dodajte i na ostale stranice u mapi (umjesto elementa `h1`).
13. Na stranicu **index.html** dodajte element `img` ispod naslova kojim ćete prikazati sliku **srce.jpg** iz mape **C201/vjezbe/vjezba2/slike**. Postavite odgovarajući tekst u atribut `alt`.
14. Na preostale stranice (**kontakt.html**, **popis.html, raspored.html**) dodajte slike po želji iz mape **C201/vjezbe/vjezba2/slike**. Ne zaboravite postaviti odgovarajući proizvoljni tekst u atribut `alt`.

7. Multimedijalne datoteke

Po završetku ovog poglavlja polaznik će moći:

- *dodati audio datoteku na web-stranicu*
- *dodati video datoteku na web-stranicu.*

Video i audio datoteke mogu se na *web*-stranice uključiti ili izravno, uporabom HTML-elemenata, ili na način da ih se prethodno postavi na neki servis namijenjen dijeljenju videa (Youtube) ili zvuka (SoundCloud) te da se od tog servisa preuzme kôd koji će se uključiti direktno u HTML-kôd *web*-stranice.

7.1. Dodavanje audio datoteke – *audio*, *source*

Dodavanje audio datoteke putem nekog od servisa namijenjenog dijeljenju audio datoteka prilično je jednostavno jer će takav servis, nakon postavljanja audio datoteke korisniku omogućiti kopiranje HTML-kôda koji se jednostavno umetne u HTML-datoteku.

HTML5 je uveo novi element `<audio>` koji omogućava dodavanje audio datoteka na *web*-stranice. U slučaju da preglednik ne podržava postavljeni audio format, prikazat će što god se nalazi unutar `<audio>` elementa.

Element <audio> ima sljedeće atribute:

- **Src** – označava put do audio datoteke
- **Controls** – postavlja kontrole na audio player. Ako se atribut ne navede, kontrole neće biti prikazane.
- **Autoplay** – u slučaju da se ovaj atribut navede, audio datoteka će se automatski pokrenuti. Smatra se dobrom praksom da se audio datoteka pokreće tek na zahtjev korisnika.
- **Preload** – u slučaju da nije postavljen atribut *autoplay* ovaj će atribut odrediti hoće li se audio datoteka automatski preuzeti na korisničko računalo ili ne. Atribut može imati vrijednost:
    - **None** – preglednik neće automatski preuzimati datoteku
    - **Auto** – preglednik će preuzeti datoteku na korisničko računalo čim se učita *web*-stranica
    - **Metadata** – preglednik će preuzeti osnovne informacije o datoteci (veličina, ime, trajanje…)
- **Loop** – određuje hoće li se ili neće po završetku audio datoteke ona ponovno pokrenuti..

Zanimljivosti i napomene

Za razliku od <video> elementa, <audio> element nije stekao veliku popularnost. Jedan od razloga mogao bi biti i to što su prvi preglednici koji su implementirali ovaj element imali problema s kvalitetom zvuka prilikom reprodukcije audio datoteka.



Zanimljivosti i napomene

Većina novih preglednika podržava format MP3, pa ovisno o ciljanoj publici nije nužno pripremati datoteku u više formata.

Primjer:

```
<h1>Audio datoteka</h1>
<p>Poslušajte ovu pjesmu! </p>
<audio src="audio/FreeMP3.mp3" controls autoplay>
Ovaj preglednik ne podržava ovu vrstu audio datoteke.
</audio>
```

Budući da neki preglednici ne podržavaju određene vrste audio datoteka, moguće je pripremiti više formata iste audio datoteke i objaviti ih na *web*-stranici, kako bi ona vrsta koju preglednik podržava bila dostupna korisniku. Ovo je moguće napraviti korištenjem HTML-elementa `<source>`.

Primjer:

```
<h1>Audio datoteka</h1>
<p>Poslušajte ovu pjesmu!</p>
<audio controls autoplay>
     <source src="audio/FreeMP3.mp3" />
<source src="audio/FreeMP3.ogg" />
Ovaj preglednik ne podržava ovu vrstu audio datoteke.
</audio>
```

7.2. Dodavanje video datoteke – *video*, *source*

Kao i s audio datotekama, dodavanje video datoteke putem nekog od servisa namijenjenog dijeljenju video datoteka prilično je jednostavno jer će takav servis, nakon postavljanja audio datoteke, korisniku omogućiti kopiranje HTML-kôda koji se jednostavno umetne u HTML-datoteku.

HTML-element `<video>` omogućava dodavanje video datoteka na *web*-stranice. U slučaju da preglednik ne podržava postavljeni video format, prikazat će što god se nalazi unutar `<video>` elementa.

Element `<video>` ima iste atribute kao i `<audio>` te neke dodatne. Dodatni su atributi:

- **width, height** – omogućava kontrolu veličine videa u pikselima. Moguće je dodijeliti oba atributa, ili samo jedan, ali je kao i kod slika potrebno paziti da se video ne deformira korištenjem oba atributa istovremeno.
- **poster** – omogućava da se tijekom učitavanja videa, na mjestu videa pojavi zadana slika.
  `poster="slike/eduroam.png"`

Primjer:

```
<h1>eduroam - studentska razmjena</h1>
<p>Kako jednostavno pristupiti Internetu kad se
nalazite na studentskoj razmjeni.</p>
<video src="video/eduroam-studentskarazmjena.mp4"
               width="600"
              preload
              controls
              autoplay>
     eduroam - besplatno, jednostavno i brzo na
     Internet u inozemstvu
</video>
```



Budući da neki preglednici ne podržavaju određene vrste video datoteka, moguće je pripremiti više formata iste video datoteke i objaviti ih na *web*-stranici, kako bi ona vrsta koju preglednik podržava bila dostupna korisniku. Ovo je moguće napraviti korištenjem HTML-elementa `<source>`.

Atributi ovog elementa su:

- **Src** – putanja do datoteke
- **Type** – atribut kojim se identificira format video datoteke. Ovaj se atribut koristi kako preglednik ne bi započinjao preuzimanje datoteke da sazna o kojem se formatu radi.
- **Codecs** – atribut kojim se označava vrsta softvera za dekodiranje videa koja se koristila pri izradi videa.

Primjer:

```
<h1>eduroam - studentska razmjena</h1>
```

```
<p>Kako jednostavno pristupiti Internetu kad se
nalazite na studentskoj razmjeni.</p>

<video width="600"
      preload
      controls
      autoplay>
      <source src="video/eduroam-
      studentskarazmjena.mp4"
      type='video/mp4;codecs="avc1.42E01E, mp4a.40.2"'
      />
      <source src="video/eduroam-
      studentskarazmjena.webm"
      type='video/webm;codecs="vp8, vorbis"' />
eduroam - besplatno, jednostavno i brzo na Internet u
inozemstvu
</video>
```

8. Tablice

Po završetku ovog poglavlja polaznik će moći:

- *izraditi jednostavne tablice*
- *izraditi tablice sa složenijom strukturom, spojenim recima ili spojenim stupcima*
- *dodati naslov tablici.*

Tablice omogućuju prikaz informacija na dvije osi. Koriste se za prikaz financijskih izračuna, rasporeda, različitih rezultata, statističkih podataka…

8.1. Struktura tablice – *table, tr i td* elementi

Za izradu tablice koristi se element `<table>`. Sadržaj u tablici ispisuje se red po red. Svaki red definiran je elementom `<tr>`. TR označava redak tablice (eng. *table row*). Svaka ćelija u retku definira se elementom `<td>`. TD označava sadržaj ćelije (eng. *table data*).

Neki preglednici automatski prikazuju i linije oko tablice, dok drugi ne.

Primjer.

```
<table>
<tr>
<td>Uvod u HTML</td>
<td>12</td>
</tr>
<tr>
<td>Uvod u PHP i MySQL</td>
<td>25</td>
</tr>
<tr>
<td>Uvod u SQL</td>
<td>20</td>
</tr>
</table>
```

Trajanje tečajeva

Uvod u HTML	12
Uvod u PHP i MySQL	25
Uvod u SQL	20

8.2. Elementi zaglavlja i podnožja – th, thead, tbody i tfoot

Elementi `<th>`, `<thead>`, `<tbody>` i `<tfoot>` koriste se kako bi se olakšao prikaz sadržaja korisnicima koji upotrebljavaju čitače za pregledavanje *web*-stranica, kako bi se poboljšalo indeksiranje stranica od strane tražilica, a također olakšavaju kontrolu nad izgledom tablice kada se koristi CSS.

Zanimljivosti i napomene

Ako je u tablici potrebno navesti praznu ćeliju, to se radi na način da se svejedno navede element <td> ili <th>. U slučaju da se ne navede prazan element, tablica će se krivo prikazati.

Korištenjem elementa <th> tablicama je moguće dodati i naslovni redak. TH označava naslov retka ili stupca (eng. *table heading*). Preglednici obično prikazuju ovaj element podebljano i centrirano u ćeliji pa je već i na prvi pogled vidljiva njegova svrha.

Primjer:

```
<table>
<tr>
<th>Naziv tečaja</th>
<th>Trajanje</th>
</tr>
<tr>
<td>Uvod u HTML</td>
<td>12</td>
</tr>
<tr>
<td>Uvod u PHP i MySQL</td>
<td>25</td>
</tr>
<tr>
<td>Uvod u SQL</td>
<td>20</td>
</tr>
</table>
```

Trajanje tečajeva

Naziv tečaja	Trajanje
Uvod u HTML	12
Uvod u PHP i MySQL	25
Uvod u SQL	20

Zanimljivosti i napomene

Zaglavlje i podnožje tablice su različiti elementi kako bi se vrlo duge tablice mogle printati ili prikazati na način da je zaglavlje i podnožje uvijek vidljivo. Ovo još nije sustavno implementirano u preglednicima.

Element <thead> označava zaglavlje tablice (eng. *table head*), element <tbody> sadržaj tablice (engl. *table body*), a element <tfoot> podnožje stranice (eng. *table footer*). Ovi se elementi neće prikazivati na drugačiji način od ostalih elemenata tablice, ali se koriste prilikom definiranja dizajna tablice.

Primjer:

```
<table>
<thead>
<tr>
<th>Naziv tečaja</th>
<th>Trajanje</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uvod u HTML</td>
<td>12</td>
</tr>
<tr>
<td>Uvod u PHP i MySQL</td>
<td>25</td>
</tr>
<tr>
<td>Uvod u SQL</td>
<td>20</td>
</tr>
</tbody>
<tfoot>
<tr>
                      <td>Ukupno trajanje svih
                      tečajeva:</td>
<td>57</td>
</tr>
</tfoot>
</table>
```

Trajanje tečajeva

Naziv tečaja	Trajanje
Uvod u HTML	12
Uvod u PHP i MySQL	25
Uvod u SQL	20
Ukupno trajanje svih tečajeva:	57

8.3. Spajanje ćelija – *colspan i rowspan*

Ponekad je potrebno spojiti ćelije u recima ili u stupcima kako bi se proširio prostor za sadržaj, ili kako bi se uredila hijerarhijska struktura tablice. To se može postići korištenjem atributa `colspan` za spajanje više ćelija u retku ili `rowspan` za spajanje više ćelija u stupcu. Ovi se atributi mogu dodati elementima `<th>` i `<td>`.

`Colspan` i `rowspan` su atributi kojima se dodjeljuje broj ćelija koje se želi spojiti u jednu. Važno je voditi računa o tome da će neke ćelije na ovaj način "nestati" iz kôda. Npr., ako redak ima tri ćelije, u kôdu se

nalaze tri elementa `<td>`. Ako se dvije ćelije spoje u jednu, u retku će ostati dvije ćelije (jedna duža i jedna kraća), pa će shodno tome u kôdu biti i dva `<td>` elementa, a ne više tri.

Primjer:

```
<table>
      <thead>
            <th></th>
            <th colspan="2">Ime tablice kroz dva
            stupca</th>
      </thead>
      <tbody>
<tr>
<td rowspan="2">Stupac 1</td>
<td>Redak 1</td>
<td rowspan="2">Stupac 2</td>
</tr>
<tr>
<td>Redak 2</td>
</tr>
<tr>
                  <td colspan="3">Redak po cijeloj
                  dužini tablice</td>
</tr>
<tr>
<td>Ćelija u stupcu 1</td>
<td>Ćelija u stupcu 2</td>
<td>Ćelija u stupcu 2</td>
</tr>
      </tbody>
</table>
```

<table>
<tr><th></th><th colspan="2">Ime tablice kroz dva stupca</th></tr>
<tr><td rowspan="2">Stupac 1</td><td>Redak 1</td><td rowspan="2">Stupac 2</td></tr>
<tr><td>Redak 2</td></tr>
<tr><td colspan="3">Redak po cijeloj dužini tablice</td></tr>
<tr><td>Ćelija u stupcu 1</td><td>Ćelija u stupcu 2</td><td>Ćelija u stupcu 2</td></tr>
</table>

8.4. Naslov tablice – *caption*

Tablice mogu imati i nazive (eng. *caption*), baš kao i slike. Za dodavanje naslova koristi se element `<caption>` koji će se smjestiti iznad tablice, a u kôdu se dodaje iznad prvog `<tr>` elementa.

Do sada se u primjerima kao naslov tablice koristio `<h1>` ili `<h2>`, standardni element koji se koristi za naslove svojega sadržaja u HTML-u, a sada će biti prikazan primjer s elementom `<caption>`.

Elementi <h1>

Primjer:

```
<table>
<caption>Trajanje tečajeva</caption>
<tr>
<th>Naziv tečaja</th>
<th>Trajanje tečaja</th>
</tr>
<tr>
<td>Uvod u HTML</td>
<td>12</td>
</tr>
<tr>
<td>Uvod u PHP i MySQL</td>
<td>25</td>
</tr>
<tr>
<td>Uvod u SQL</td>
<td>20</td>
</tr>
</table>
```

Trajanje tečajeva

Naziv tečaja	**Trajanje tečaja**
Uvod u HTML	12
Uvod u PHP i MySQL	25
Uvod u SQL	20

8.5. Vježba: Audio i tablice

1. Otvorite postojeći HTML-dokument **index.html** u mapi **C201/vjezbe/vjezba3**.
2. Dodajte još jedan odlomak s tekstom "Poslušajte izvadak iz online tečaja Srca".
3. Dodajte element `audio` koji će omogućiti preslušavanje audio datoteke **tecaj.mp3** koja se nalazi u mapi **C201/vjezbe/vjezba3/audio**. Isprobajte radi li preslušavanje datoteke u pregledniku.
4. Otvorite postojeći HTML-dokument **raspored.html** koji se nalazi u mapi **C201/vjezbe/vjezba3**.
5. Kopirajte sadržaj tekstualne datoteke **raspored.txt** i zalijepite ga poslije zadnjeg odlomka u dokumentu (a prije elementa `hr`).
6. Strukturirajte kopirani sadržaj u tablicu koja će prikazivati raspored tečajeva, koristeći elemente `table`, `tr`, `th` i `td`. Kako bi tablica bila vidljiva postavite joj atribut `border` na vrijednost 1.

9. Uključivanje CSS-a

Po završetku ovog poglavlja polaznik će moći:

- *objasniti što je CSS i čemu služi*
- *uključiti vanjsku CSS-datoteku u HTML-datoteku*
- *dodati CSS-kôd izravno u HTML-kôd.*

CSS je jezik koji služi za oblikovanje *web*-stranica. Uz HTML, CSS je osnovna tehnologija na kojoj se temelji današnji *web*, a služi prvenstveno za oblikovanje sadržaja. Kroz CSS kôd definira se izgled *web*-stranice i svih elemenata na njoj.

9.1. O CSS-u

CSS je kratica za *Cascading Style Sheets*.

Pojam *style sheet* često se upotrebljava za datoteku koja sadrži CSS-kôd. Dakle, *style sheet* je datoteka koja definira stil, odnosno izgled *web*-stranice.

Riječ *cascading* označava kaskadnu primjenu CSS-pravila. CSS-pravilo može se napisati tako da bude primijenjeno na sve elemente ili samo na neke elemente ili tako da vrijedi samo za točno određeni element.

Prije pojave CSS-a oblikovanje izgleda *web*-stranice do određene razine bilo je moguće postići i u HTML-u. No, time se stvorio problem miješanja sadržaja i strukture s kôdom čija je jedina svrha bila prezentacija. HTML-kôd za definiranje izgleda morao se ponavljati iznova na svakom elementu i na svakoj stranici u *web*-sjedištu.

Pojavom CSS-a nastoji se riješiti taj problem. Glavna je ideja CSS-a odvajanje prezentacijskoga kôda u zasebne datoteke i njegovo definiranje pomoću jednostavnih pravila koja se mogu odnositi na više elemenata odjednom.

Trenutačna verzija CSS-a je CSS 3, koji je donio brojne novitete i poboljšanja. Ta verzija još nije konačna, nego se i dalje neprestano razvija i nadograđuje.

9.2. Uključivanje vanjske CSS-datoteke – *link* element

CSS-kôd se uobičajeno piše odvojeno od HTML-kôda, tj. u zasebnoj datoteci. Da bi se HTML-dokument povezao s CSS datotekom, koristi se HTML-element `<link>`.

Primjer:

```
<link rel="stylesheet" type="text/css"
      href="stilovi/boje.css">
```

Atribut rel mora imati vrijednost "stylesheet", a atribut type vrijednost "text/css", da bi se pomoću elementa uključio CSS. Atribut href postavlja se na putanju do CSS-datoteke koju se želi uključiti.

U ovom primjeru uključujemo datoteku ***boje.css*** koja se nalazi u mapi ***stilovi.***

Element link mora se uvijek nalaziti unutar HTML elementa head.

Ako se HTML-stranica koristi većim brojem CSS datoteka, uključit će se tako da se element link navede više puta.

Mogući su i načini pisanja CSS-kôda u samoj HTML-datoteci. Za to služi HTML-element style, u kojem se izravno mogu pisati CSS-pravila.

Primjer:

```
<style type="text/css">
      p
      {
            color: red;
      }
</style>
```

Element style uvijek se mora nalaziti unutar elementa head. Atribut type u tom se slučaju postavlja na vrijednost "type/css", a može se i izostaviti.

Drugi način na koji se CSS-kôd može izravno "ubaciti" u HTML je preko atributa style.

Primjer:

```
<p style="color:red;">
      Tko rano rani, dvije sreće grabi.
</p>
```

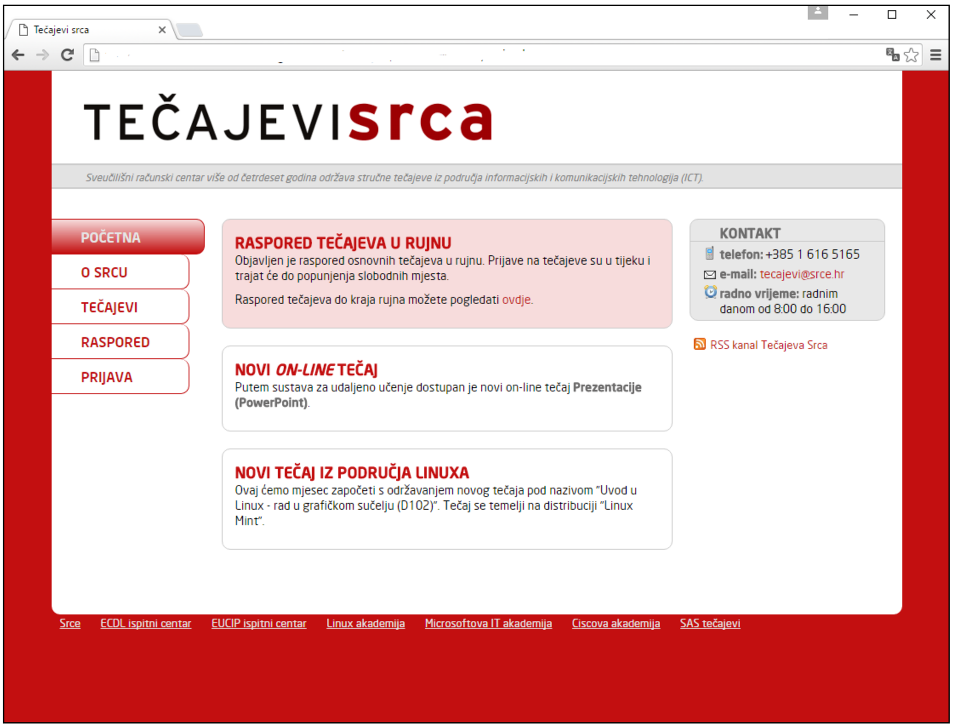
Atribut style može se staviti na gotovo svaki HTML-element. U ovom se slučaju pišu samo CSS-deklaracije, bez selektora, a te će se deklaracije primijeniti samo na taj HTML-element.

No, iako se ponekad može činiti praktičnim, miješanje se CSS-a i HTML-a u istoj datoteci ne preporuča. Najbolji pristup je pisanje CSS-kôda u zasebnoj datoteci. Time se ostvaruje ponovna iskoristivost kôda i sažetost – određeno CSS-pravilo dovoljno je napisati samo jednom, a ono će vrijediti u svim HTML-dokumentima. Također, kada je potrebno nešto promijeniti u nekom CSS-pravilu, dovoljno je to napraviti na jednome mjestu, a ne u svakoj HTML-datoteci zasebno.

Primjer stranice bez uključenog CSS-a:



Primjer stranice s uključenim SS-om:



9.3. Uključivanje ikone stranice - *favicon*

Ikona stranice (tzv. *favicon*) je mala sličica koja se prikazuje u zaglavlju preglednika kako bi se stranica vizualno razlikovala od drugih stranica otvorenih u pregledniku.

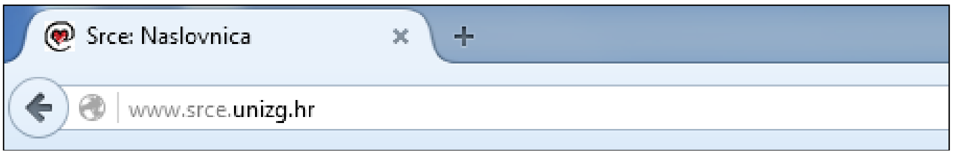
Ikona stranice se, kao i CSS-datoteke, uključuje pomoću elementa `<link>`.

Primjer:

```
<link rel="icon "href="favicon.ico">
```

Atribut `rel` mora imati vrijednost `icon`, a pomoću atributa `href` postavlja se putanja do datoteke.

Za ikonu stranice najčešće se koristi .ico datoteka, iako su podržani i drugi formati (npr. .png i .gif).



10. Strukturalni elementi

Po završetku ovog poglavlja polaznik će moći:

- *opisati što su strukturalni elementi i čemu služe*
- *objasniti čemu služe <div> i <span> elementi te kako se razlikuju od ostalih strukturalnih elemenata*
- *napisati osnovnu strukturu HTML-stranice.*

Kako je ranije navedeno, HTML dokument ima određenu strukturu. HTML-dokument tako može imati naslove, odlomke, slike, tablice i slično. Osim ovoga, HTML-dokument može imati i elemente koji definiraju strukturu same *web*-stranice kao što su navigacija, zaglavlje, podnožje, članak... Ovi elementi nazivaju se strukturalnim elementima. Strukturalni elementi novi su HTML-elementi koji su se pojavili s inačicom HTML5 i služe prvenstveno za slaganje osnovnih elemenata stranice u određenu strukturu, tj. slaganje *layouta* stranice.

Ranije, dok CSS još nije bio dovoljno razvijen, struktura HTML-stranice izrađivala se na način da se sav sadržaj stranice smještao u tablice koje nisu imale vidljivi rub (*border*). Korištenjem `colspan` i `rowspan` atributa mogla se izraditi bilo kakva željena struktura tablice, te bi sadržaj umetnut u nju zadržavao željenu strukturu. Ovo je razvojem CSS-a izgubilo svrhu te se danas smatra izrazito lošim načinom izrade *web*-stranica. Dugo nakon toga, autori *web*-stranica koristili su <div> elemente kako bi grupirali elemente na stranici.

S dolaskom HTML5 standarda uveden je novi niz elemenata koji omogućuju podjelu sadržaja i dijelova stranice. Već sama imena tih elemenata na neki način pojašnjavaju čemu koji element služi.

Dodatni smisao novih elemenata je i opisivanje strukture stranice koje je korisno i za prilagodbu *web*-stranica za osobe s invaliditetom s obzirom na to da će čitači na taj način lakše omogućiti preskakanje zaglavlja prilikom pregledavanja *web*-stranica, tj. skakanje izravno na sam sadržaj i slično. Osobi koja koristi čitač to znatno skraćuje vrijeme pronalaska važnoga sadržaja.

Također, korisni su i za bolju pronalažljivost stranica od strane tražilica koje će lakše pronaći relevantan sadržaj preskačući npr. zaglavlje ili podnožje stranice i pregledavajući sadržaj u člancima (element *article*).

10.1. Elementi za grupiranje - *div* i *span*

HTML-element `<div>` je blok element koji omogućuje grupiranje više elemenata unutar jednoga bloka.

Moguće je izraditi jedan `<div>` element u kojem će se pojavljivati više elemenata koji logički spadaju na isto mjesto, npr. komentari korisnika, slike i slično. Zatim se na tom `<div>` elementu može primjeniti točno određeni CSS koji će omogućiti da svi elementi unutar toga `<div>` elementa izgledaju jednako. To bi značilo da će npr. svi komentari na tome mjestu korisniku izgledati kao dio sekcije za komentare ili da će sve slike izgledati kao galerija.

Bez implementacije CSS-a, sadržaj `<div>` elementa u pregledniku neće biti prezentiran na neki poseban način osim što će obavezno započeti u novom redu, s obzirom na to da je riječ o blok elementu.

Element `<div>` također se koristi i za grupiranje drugih strukturalnih elemenata na stranici.

Primjer:

```
<div class="srednji">
      <div class="clanak istaknuto">
            <h2>Raspored tečajeva u rujnu</h2>
            <p>Objavljen je raspored osnovnih tečajeva
            u rujnu. Prijave na tečajeve su u tijeku i
            trajat će do popunjenja slobodnih
            mjesta.</p>
            <p>Raspored tečajeva do kraja rujna možete
            pogledati <a
            href="raspored.html">ovdje</a>.</p>
      </div>
      <div class="clanak">
            <h2>Novi <em>on-line</em> tečaj</h2>
            <p>Putem sustava za udaljeno učenje
            dostupan je novi on-line tečaj
            <strong>Prezentacije (PowerPoint
            2010)</strong>.</p>
      </div>
</div>
```

HTML-element `<span>` je generički *inline* element koji omogućuje označavanje dijela teksta na koji se želi primijeniti specifično CSS-svojstvo, tj. označavanje dijela sadržaja u slučaju da nijedan drugi semantički element nije iskoristiv te za grupiranje drugih *inline* elemenata. Ako se elementom `<span>` označi neki tekst, bez primjene nekog CSS-kôda, taj će se tekst prikazati kao sav ostali tekst na stranici.

Primjer:

```
<p><span class="telefon">telefon:</span> +385 1 616
5165</p>
```

10.2. Zaglavlje i podnožje stranice – *header* i *footer*

Elementi koji se koriste za definiranje zaglavlja `<header>` i podnožja `<footer>` mogu se koristiti za:

- Definiranje zaglavlja i podnožja za cijelu *web*-stranicu
- Definiranje zaglavlja i podnožja nekog članka `<article>` ili sekcije, tj. dijela *web*-stranice `<section>`.

U navedenim elementima mogu se grupirati svi ostali elementi koji bi bili dio zaglavlja ili podnožja neke stranice, npr. ime *web*-stranice, navigacija, informacije o autorskim pravima i slično.

U slučaju da se zaglavlje koristi u članku, u njemu može biti naslov članka ili nešto slično.

Primjer:

```
<header>
      <img src="slike/logo.png" alt="Tečajevi srca" />
      <p> Sveučilišni računski centar više od
      četrdeset godina održava stručne tečajeve iz
      područja informacijskih i komunikacijskih
      tehnologija (ICT).
      </p>
<div class="navigacija">
            <a href="index.html"
            class="selected">Početna</a>
<a href="srce.html">O Srcu</a>
<a href="tecajevi.html">Tečajevi</a>
            <a href="raspored.html">Raspored</a>
            <a href="prijava.html">Prijava</a>
      </div>
</header>
```

Primjer:

```
<footer>
      <a href="#">Srce</a>
<a href="#">Linux akademija</a>
<a href="#">Ciscova akademija</a>
</footer>
```

10.3. Navigacija – element *nav*

Element <nav> koristi se za držanje navigacije na jednome mjestu, npr. svu glavnu navigaciju dobro je smjestiti unutar elementa <nav>.

Primjer:

```
<nav>
     <a href="index.html"
     class="selected">Početna</a>
     <a href="srce.html">O Srcu</a>
     <a href="tecajevi.html">Tečajevi</a>
     <a href="raspored.html">Raspored</a>
     <a href="prijava.html">Prijava</a>
</nav>
```

10.4. Sadržaj – elementi *article* i *aside*

Element <article> koristi se kao svojevrsno spremište za bilo koji dio stranice koji može biti samostalan ili se može preuzeti i distribuirati kao cjelina. To može biti članak, samostalni tekst, forum, galerija…

Unutar elementa <article> smješta se jedan samostalni dio sadržaja. Ako je riječ o komentarima, oni se mogu smjestiti unutar elementa <article> koji može biti ugniježđen unutar drugoga <article> elementa u kojemu se nalazi članak.

Primjer:

```
<article>
<header>
<h1 id="vrh">Osnovni tečajevi Srca</h1>
     </header>
     <figure>
          <img src="slike/osnovni_tecajevi.png"
          alt="Polaznici tečajeva" title="Polaznici
          tečajeva u prostorijama Sveučilišnog
          računskog centra"/>
     <br>
          <figcaption>Polaznici na tečaju u
          Srcu</figcaption>
</figure>
     <p>Osnovni tečajevi Srca namijenjeni su svima
     zainteresiranima za stjecanje osnovnih znanja o
     uporabi informacijsko-komunikacijskih
     tehnologija. Osnovni se tečajevi održavaju u
     učionicama (prema rasporedu održavanja), ali i u
     online obliku (putem sustava za udaljeno
     učenje).</p>
</article>
```

Element <aside> može se koristiti na dva načina, ovisno o tome gdje je smješten:

- Ako je smješten unutar elementa <article> trebao bi sadržavati informacije koje su vezane isključivo uz sadržaj u tom elementu, a ne informacije koje se odnose na cjelokupan sadržaj stranice, npr. rječnik vezan uz članak.

- Ako je smješten van elementa <article> onda može služiti kao spremnik za informacije koje se tiču cijele stranice, na primjer može sadržavati linkove na druge djelove stranice, okvire za umetanje vezanih informacija s društvenih mreža, popis novih članaka i slično.

Primjer:

```
<article>
     <header>
          <h1 id="vrh">Osnovni tečajevi Srca</h1>
     </header>
     <figure>
          <img src="slike/osnovni_tecajevi.png"
          alt="Polaznici tečajeva" title="Polaznici
          tečajeva u prostorijama Sveučilišnog
          računskog centra"/>
     <br>
          <figcaption>Polaznici na tečaju u
          Srcu</figcaption>
     </figure>
     <p>Osnovni tečajevi Srca namijenjeni su svima
     zainteresiranima za stjecanje osnovnih znanja o
     uporabi informacijsko-komunikacijskih
     tehnologija. Osnovni se tečajevi održavaju u
     učionicama (prema rasporedu održavanja), ali i u
     online obliku (putem sustava za udaljeno
     učenje).
     </p>
     <aside>
          <a href="#">Raspored održavanja
          tečajeva</a>
          <a href="#">Prijavite se na tečaj</a>
          <a href="#">Prijava u sustav za udaljeno
          učenje</a>
     </aside>
</article>
```

10.5. Sekcija – element *section*

Element `<section>` omogućuje grupiranje međusobno povezanoga sadržaja u sekcije. Uobičajeno je da svaka sekcija ima vlastito zaglavlje.

Na jednoj *web*-stranici može biti više sekcija. Svaka sekcija može imati nekoliko `<article>` elemenata koji su međusobno logički povezani, tj. imaju zajedničku temu ili svrhu.

Trebalo bi se izbjegavati smještanje cjelokupnoga sadržaja *web*-stranice unutar jednoga `<section>` elementa. Za to je bolje koristiti element `<div>`.

Primjer:

```
<section>
	<header>
		<h1 id="vrh">Popis tečajeva:</h1>
	</header>
	<ul>
		<li><a href="#sql">Uvod u SQL</a></li>
		<li><a href="#ruby">Uvod u programski
		jezik Ruby</a></li>
		<li><a href="#PHP">Uvod u PHP i
		MySQL</a></li>
	</ul>
</section>
<section>
	<article>
		<header>
			<h2 id="sql">Uvod u SQL</h2>
		</header>
		<p>D300</p>
		<p>Objekti, relacije i baze podataka;
		kompleksne selekcije; operacije nad
		relacijama; elementi teorije
		normalizacije; oblikovanje relacijske baze
		podataka.</p>
	</article>
	<article>
		<header>
			<h2 id="ruby">Uvod u programski jezik
			Ruby</h2>
		</header>
		<p>D400</p>
		<p>Osnove jezika Ruby, izrazi, klase i
		objekti, rad s datotekama, upravljanje
		greškama, moduli, testiranje programskog
		koda.</p>
	</article>
<article>
		<header>
			<h2 id="PHP">Uvod u PHP i
		MySQL</h2></header>
		<p>C200</p>
```

```
          <p>Model klijent-poslužitelj, varijable,
          operatori, uvjetne strukture, polja,
          petlje, funkcije, ugrađene funkcije PHP-a,
          obrasci, prijenos podataka između skripti,
          rad s datotekama, slanje poruke
          elektroničke pošte, rad s bazom podataka
          MySQL, sjednice, autentikacija
          korisnika.</p></article>
</section>
<p>Povratak na <a href="#vrh">vrh</a>.</p>
```

10.6. Vježba: Strukturalni elementi i CSS

1. Otvorite postojeći HTML-dokument **index.html** u mapi **C201/vjezbe/vjezba4**.

2. Logotip u zaglavlju i popis s navigacijskim linkovima postavite unutar strukturalnoga elementa `header`.

3. Popis s navigacijskim linkovima postavite unutar strukturalnoga elementa `nav`.

4. Obrišite `hr` element iz podnožja stranice i postavite strukturalni element `footer` oko teksta "© Sveučilišni računski centar – Srce".

5. Glavni dio sadržaja stranice (uključujući i naslov) stavite u strukturalni element `article`.

6. Ponovite ovaj postupak i na ostalim stranicama u mapi (**kontakt.html**, **popis.html, raspored.html**).

7. Izgled stranica u pregledniku nije se previše promijenio. Međutim, korištenjem strukturalnih elemenata namjena pojedinih dijelova stranice postaje poznata, a moguće je i napisati CSS-pravila koja se odnose na pojedini strukturalni element. Uključite CSS-datoteku **izgled.css** koja se nalazi u mapi **C201/vjezbe/vjezba4/css** na sve stranice (**index.html, kontakt.html**, **popis.html, raspored.html**). Provjerite njihov izgled u pregledniku.

8. Uključite ikonu stranice favicon.ico koja se nalazi u istoj mapi. Postavite je na sve stranice u mapi (**index.html, kontakt.html**, **popis.html, raspored.html**).

11. Obrasci

Po završetku ovog poglavlja polaznik će moći:

- *objasniti što su i kako funkcioniraju obrasci*
- *koristiti element form i njegove atribute*
- *koristiti jednostavne elemente za unos teksta*
- *koristiti elemente za odabir vrijednosti*
- *koristiti dugme za slanje podataka*
- *odabrati ispravan element za određeni tip podatka*
- *koristiti validaciju ugrađenu u preglednik*
- *grupirati elemente obrasca.*

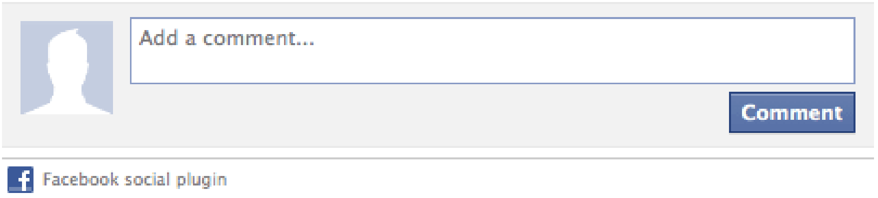
Obrasci na *webu* služe kako bi korisnici mogli upisati neke podatke u preglednik i poslati ih poslužitelju. Ovi podaci mogu biti tekst koji se upisuje u polje za pretraživanje, ili pak podaci koje korisnik upisuje u polje za unos komentara na *web* stranici, *online* anketu, ili u sučelje *web* aplikacije.

Glavna značajka obrazaca na *webu* je dugme čijim se pritiskom uneseni podaci šalju poslužitelju. Poslužitelj obrađuje podatke i prikazuje korisniku povratnu informaciju – rezultate pretrage, ili poruku o uspješnom spremanju podataka.

Najpoznatiji primjer obrasca je početna stranica tražilice Google:



Drugi poznati primjer je polje za unos komentara na Facebooku:



11.1. Element *form*

Element <`form`> je glavni HTML-element *web* obrasca. Unutar ovoga elementa nalaze se ostali elementi obrasca.

Element <`form`> ima sljedeće atribute:

- **method** – način slanja podataka. Moguće vrijednosti su
  - **get** – podaci se šalju putem URL-a
  - **post** – podaci se šalju u tijelu HTTP zahtjeva
- **action** – putanja do stranice (poslužiteljske skripte) koja obrađuje podatke. Putanja može biti apsolutna ili relativna
- **enctype** – način enkodiranja podataka

Primjer:

```
<form method="post" action="spremiPodatke.php">

</form>
```

11.2. Polje za unos teksta

Najčešće korišteni element obrazaca na *webu* je polje za unos teksta. Većina elemenata obrazaca, pa tako i polje za unos teksta dobiva se pomoću HTML-elementa `<input>`.

Ovo su osnovni atributi koji se koriste kod polja za unos teksta:

- **type** – postavljen na vrijednost `text`
- **name** – naziv polja koji se šalje zajedno s unesenim podacima
- **value** – početni tekst upisan u polju
- **maxlength** – maksimalni dozvoljeni broj znakova
- **placeholder** – tekst koji daje primjer unosa.

Primjer:

```
<input type="text" name="ime">
```



Kod ovakvoga polja za unos teksta moguć je unos samo jedne linije teksta.

Pomoću atributa `maxlength` moguće je ograničiti maksimalan broj znakova koji se mogu unijeti u polje.

Atribut `placeholder` omogućuje prikaz teksta koji će korisniku dodatno pojasniti namjenu polja. Najčešće se radi o tekstu koji daje primjer

unosa, ili dozvoljeni / preporučeni format unosa. Klikom na polje ovaj će tekst nestati, a korisnik može nesmetano unijeti željeni tekst.

Primjer:

```
<input type="text" name="adresa"
 placeholder="Naziv ulice i kućni broj">
```



11.3. Dugme (*button* i *input* element)

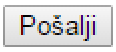
Obavezni element svakoga obrasca je dugme pomoću kojega se podaci uneseni u obrazac šalju poslužitelju. Dugme za slanje obrasca može se dobiti pomoću elementa `input` s atributom `type` postavljenim na vrijednost `submit`, ili pomoću elementa `button`.

Ovo su osnovni atributi koji se koriste kad se element `input` koristi kao dugme za slanje obrasca:

- **type** – postavljen na vrijednost `submit`
- **name** – naziv polja koji se šalje zajedno s podacima
- **value** – tekst prikazan na dugmetu (i vrijednost koja se šalje ako je gumb pritisnut).

Primjer:

```
<input type="submit" value="Pošalji">
```



Glavni razlog za upotrebu elementa `button` umjesto elementa `input` je to što je unutar elementa `button` moguće umetnuti druge HTML-elemente te na taj način dobiti formatirani tekst ili sliku.

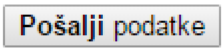
Također, kod elementa `button` vrijednost koja se šalje ako je dugme pritisnuto može biti različita od teksta prikazanog u dugmetu. Ova vrijednost je bitna ako obrazac ima više od jednog gumba za slanje pa je potrebno utvrditi koje dugme je pritisnuto.

Kod elementa `button` atributi se koriste ovako:

- **type** – postavljen na vrijednost `submit` (može se i izostaviti)
- **name** – naziv polja koji se šalje zajedno s podacima
- **value** – vrijednost koja se šalje ako je gumb pritisnut

Primjer:

```
<button>
<strong>Pošalji</strong>podatke
</button>
```



U radu s obrascima susreće se i druga vrsta dugmeta – dugme za čišćenje obrasca, odnosno povratak obrasca na početne vrijednosti. Ovo dugme se može dobiti pomoću elementa `input` ili pomoću elementa `button`, kod kojih je atribut `type` postavljen na vrijednost `reset`.

Primjer:

```
<input type="reset" value="Očisti">
```

Ili:

```
<button type="reset">Očisti</button>
```



Na kraju, moguća je i treća vrsta dugmeta, koju se također može dobiti pomoću elementa `input` ili pomoću elementa `button`, kod kojih je atribut `type` postavljen na vrijednost `button`. Ova vrsta dugmeta nema nikakvu ugrađenu funkcionalnost, već je ponašanje dugmeta potrebno definirati pomoću *JavaScript*a.

Primjer:

```
<input type="button" value="Provjeri">
```

Ili:

```
<button type="button">Provjeri</button>
```

11.4. Polje za unos lozinke

Polje za unos lozinke također je ostvareno pomoću HTML-elementa `<input>`. Koristi se kada je potrebno da korisnik upiše lozinku ili drugi povjerljivi podatak.

Osnovni atributi koji se koriste kod ovog polja su sljedeći:

- **type** – postavljen na vrijednost `password`
- **name** – naziv polja koji se šalje zajedno s unesenim podacima

Primjer:

```
<input type="password" name="lozinka">
```



Posebnost polja za unos lozinke je u tome što, radi sigurnosti, znakovi koje korisnik upisuje nisu vidljivi na ekranu.

11.5. Polje za unos broja

Polje za unos broja ostvareno je pomoću HTML-elementa `<input>` s atributom `type` postavljenim na vrijednost `number`. Koristi se kada korisnik treba upisati brojčani podatak.

Osnovni atributi koji se koriste kod ovog polja su sljedeći:

- **type** – postavljen na vrijednost `number`
- **name** – naziv polja koji se šalje zajedno s unesenim podacima
- **value** – početna vrijednost upisana u polju
- **min** – minimalna dozvoljena vrijednost
- **max** – maksimalna dozvoljena vrijednost
- **step** – korak između dozvoljenih vrijednosti (defaultno 1).

Primjer:

```
<input type="number" name="dob" min="1" max="100">
```



U ovo polje moguće je upisati samo brojke (i eventualno decimalni separator). Uz polje se pojavljuju strelice kojima je moguće povećavati ili smanjivati vrijednost za korak postavljen u atributu `step`.

Pomoću atributa `min` i `max` moguće je ograničiti raspon vrijednosti koje je dozvoljeno unijeti. U gornjem će primjeru dozvoljene vrijednosti biti 1, 2, 3 do 100.

Ako se atribut `step` postavi na 2, dozvoljene vrijednosti bit će 1, 3, 5 i tako dalje.

Ako se atribut step postavi na 0.1, dozvoljene vrijednosti bit će 1, 1.1, 1.2, 1.3, i tako dalje.

Kod mobilnih preglednika, prilikom unosa u ovo polje nudi se numerička tipkovnica, što znatno olakšava unos u polje.

Zanimljivosti i napomene

Polja za unos broja, e-mail adrese, URL-a, telefonskog broja, datuma i polje za pretragu dio su HTML5 standarda. Ova polja za unos (s iznimkom datumskog) podržana su u svim modernim preglednicima, a u starim preglednicima ponašaju se kao obično polje za unos teksta `<input type="text">`.

11.6. Polja za unos e-mail adrese, URL-a i telefonskog broja

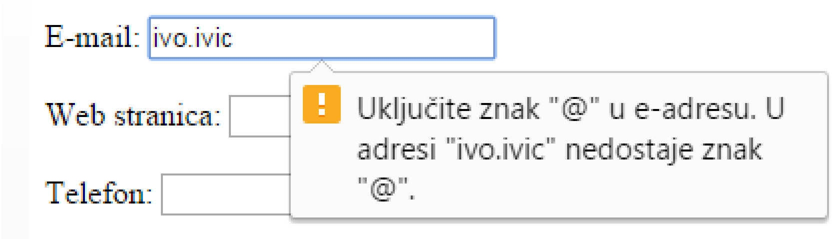
Polja za unos e-mail adrese, URL-a i telefonskog broja su, slično polju za unos broja, elementi za unos specijalne vrste podataka.

Osnovni atributi koji se koriste kod ovih polja su sljedeći:

- **type** – postavljen na vrijednost `email`, `url` ili `tel`
- **name** – naziv polja koji se šalje zajedno s unesenim podacima
- **value** – početni tekst upisan u polju
- **maxlength** – maksimalni dozvoljeni broj znakova
- **placeholder** – tekst koji daje primjer unosa

Primjer:

```
<input type="email" name="emailAdresa">
<input type="url" name="webAdresa">
<input type="tel" name="telefon">
```



Iako ova polja izgledaju identično kao i obično polje za unos teksta, ona znatno olakšavaju unos ovih specifičnih podataka.

Kod polja za unos e-mail adrese i URL-a bit će dozvoljen unos samo ispravne e-mail adrese, odnosno URL-a – ovisno o tipu polja. Ako podatak ne odgovara definiranome formatu, preglednik prikazuje poruku o pogrešci.

Na mobilnim uređajima ponuđena je prilagođena tipkovnica koja nudi samo znakove koji se koriste za unos ovih podataka.

11.7. Polje za unos datuma

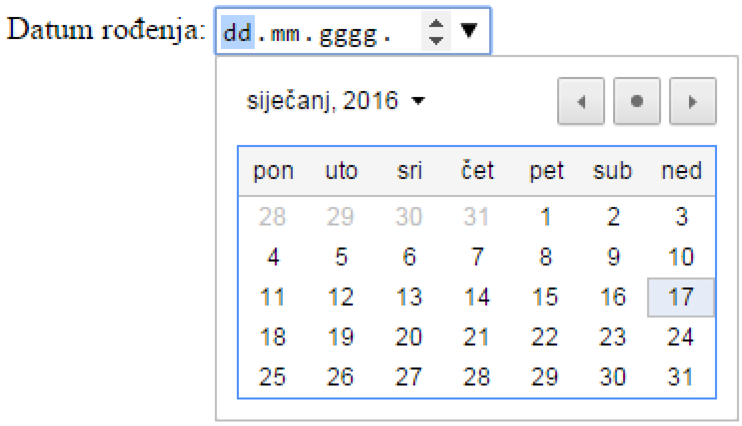
Polje za unos datuma također je ostvareno pomoću HTML-elementa `<input>`. Koristi se kada korisnik treba unijeti, odnosno odabrati ispravan datum.

Osnovni atributi koji se koriste kod ovog polja su sljedeći:

- **type** – postavljen na vrijednost `date`
- **name** – naziv polja koji se šalje zajedno s unesenim podacima
- **value** – početna vrijednost upisana u polju
- **min** – minimalna dozvoljena vrijednost (u formatu yyyy-mm-dd)
- **max** – maksimalna dozvoljena vrijednost (u formatu yyyy-mm-dd)

Primjer:

```
<input type="date" name="datum" max="2050-12-31">
```



Klikom na polje za unos datuma pojavljuje se kalendarska kontrola za odabir datuma. Također, preglednik ne dozvoljava unos neispravnoga datuma, već u tom slučaju prikazuje poruku o pogrešci.

Ovo polje zasada nije implementirano u preglednicima *Firefox* i *InternetExplorer*.

Zanimljivosti i napomene

Datumsko polje trenutačno nije podržano u svim preglednicima (podržavaju ga *GoogleChrome, Microsoft Edge, Opera, Android Browser* i mobilni *Safari*.

11.8. Polje za pretragu

Za unos teksta za pretragu može se, umjesto običnoga polja za unos teksta, koristiti posebno polje za pretragu.

Osnovni atributi koji se koriste kod ovog polja su sljedeći:

- **type** – postavljen na vrijednost `search`
- **name** – naziv polja koji se šalje zajedno s unesenim podacima
- **maxlength** – maksimalni dozvoljeni broj znakova
- **placeholder** – tekst koji daje primjer unosa

Primjer:

```
<input type="search" name="pretraga">
```



U nekim preglednicima ovo polje dobiva specijalne funkcionalnosti – pojavljuje se križić za brisanje unesenoga teksta, te se prikazuju ranije uneseni pojmovi za pretragu.

Iako nema velike razlike u odnosu na obično polje za unos teksta, preporuča se koristiti polje za pretragu tamo gdje se radi o pretraživanju, jer se time daje semantička informacija o namjeni polja.

11.9. Dugme za odabir

Zanimljivosti i napomene

Naziv *radio button* dolazi od dugmadi na starim radio aparatima, kod kojih se pritiskom na jedno dugme automatski izbacuje i time isključuje ostalu dugmad.

Dugme za odabir (eng. *radio button*) upotrebljava se kada korisnik treba odabrati vrijednost s popisa vrijednosti (ali samo jedna vrijednost može bti odabrana).

Ovo su osnovni atributi koji se koriste kod ovog elementa:

- **type** – postavljen na vrijednost `radio`
- **name** – naziv polja koji se šalje zajedno s unesenim podacima
- **value** – vrijednost koja se šalje poslužitelju
- **checked** – postavlja se ako dugme treba biti unaprijed odabrano (*Booleov* atribut)

Primjer:

```
<input type="radio" name="spol" value="m"
checked>muški

<input type="radio" name="spol" value="f">ženski

<input type="radio" name="spol" value="n">ne želim
odgovoriti
```



Dugmad za odabir koja se nalazi na istome popisu vrijednosti treba imati istu vrijednost atributa `name`. Na taj će način biti moguće odabrati samo jednu opciju – odabir jedne opcije s popisa automatski isključuje druge opcije (ako su prethodno bile odabrane).

11.10. Kvačica za označavanje

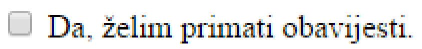
Kvačica za označavanje (eng. *checkbox*) služi da bi se uključila ili isključila neka opcija.

Ovo su osnovni atributi koji se koriste kod ovog elementa:

- **type** – postavljen na vrijednost `checkbox`
- **name** – naziv polja koji se šalje zajedno s unesenim podacima
- **value** – vrijednost koja se šalje poslužitelju (ne ispisuje se)
- **checked** – postavlja se ako kvačica treba biti unaprijed označena (*Booleov* atribut).

Primjer:

```
<input type="checkbox" name="obavijesti" value="da">
Da, želim primati obavijesti.
```



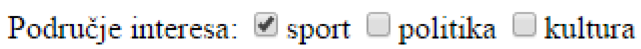
Popis ponuđenih opcija kod koje je moguće odabrati više od jedne opcije najčešće se ostvaruje pomoću popisa kvačica za označavanje (po jedna za svaku opciju).

Primjer:

```
<input type="checkbox" name="interes" value="sport"
checked>sport

<input type="checkbox" name="interes"
value="politika">politika

<input type="checkbox" name="interes"
value="kultura">kultura
```



11.11. Polje za odabir datoteke

Zanimljivosti i napomene

Da bi slanje datoteke na poslužitelj funkcioniralo, potrebno je na elementu `form` postaviti atribut `enctype` na vrijednost `multipart/form-data`.

Polje za odabir datoteke služi kako bi korisnik mogao odabrati datoteku sa svoga računala. Klikom na dugme za slanje obrasca, i datoteka će biti poslana na poslužitelj.

Ako se koristi polje za odabir datoteke nije moguće koristiti metodu *get* (putem URL-a) za slanje podataka na poslužitelj.

Ovo su osnovni atributi koji se koriste kod ovog elementa:

- **type** – postavljen na vrijednost `file`
- **name** – naziv polja koji se šalje zajedno s podacima
- **accept** – lista dozvoljenih tipova datoteka (npr. `.jpg,.png,.doc`)
- **multiple** – postavlja se ako se dopušta odabir više datoteka (*Booleov* atribut).

Primjer:

```
<input type="file" name="datoteka">
```



Izgled ovog polja, dugmeta za odabir i teksta koji se ispisuje u dugmetu, kao i dijaloga za odabir datoteke, nije moguće mijenjati već on ovisi o pregledniku i operacijskome sustavu računala.

11.12. Skriveno polje

Skriveno polje se koristi kada se želi poslati vrijednost koja neće biti vidljiva korisniku. Primjer takve vrijednosti može biti identifikator zapisa iz baze ili druge vrijednosti koje su potrebne poslužitelju, a nije ih potrebno prikazati korisniku.

U tu svrhu koristi se element `input` s atributom `type` postavljenim na vrijednost `hidden`.

Ovo su osnovni atributi koji se koriste kod ovog polja:

- **type** – postavljen na vrijednost `hidden`
- **name** – naziv polja koji se šalje zajedno s podacima
- **value** – vrijednost koja se šalje poslužitelju

Primjer:

```
<input type="hidden" name="id" value="1742">
```

Ovo polje nije vidljivo u pregledniku.

11.13. Lista za odabir

Lista za odabir omogućuje odabir jedne (ili više) ponuđenih opcija. U sličnu svrhu može se koristiti i dugmad za odabir (*radio buttons*) odnosno kvačice za označavanje (*checkboxes*), no ako se radi o većem broju ponuđenih vrijednosti, bolje je upotrijebiti listu za odabir.

Lista za odabir dobiva se pomoću HTML-elementa `select`. Njegovi atributi su sljedeći:

- **name** – naziv polja koji se šalje zajedno s unesenim podacima
- **multiple** – je li moguće odabrati više vrijednosti s liste (*Booleov* atribut)

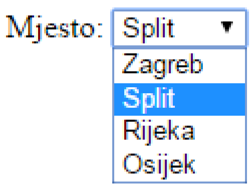
Za svaku opciju unutar liste za odabir potrebno je navesti po jedan HTML-element `option`.Njegovi atributi su sljedeći:

- **value** – vrijednost koja se šalje na poslužitelj ako je ta opcija odabrana
- **selected** – je li ova opcija unaprijed odabrana (*Booleov* atribut)

Tekst koji se ispisuje za svaku opciju piše se unutar elementa `option`

Primjer:

```
<select name="grad">
<option value="zg">Zagreb</option>
<option value="st" selected>Split</option>
<option value="ri">Rijeka</option>
<option value="os">Osijek</option>
</select>
```



Pomoću atributa `multiple` moguće je dobiti listu za odabir u kojoj se može odabrati više od jedne vrijednosti:

```
<select name="grad" multiple>
<option value="zg">Zagreb</option>
<option value="st">Split</option>
<option value="ri">Rijeka</option>
<option value="os">Osijek</option>
```

```
</select>
```



11.14. Polje za višelinijski unos teksta

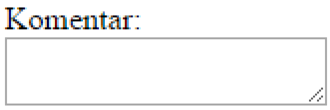
Kada je potrebno omogućiti unos većega teksta u obrazac, umjesto elementa `<input type="text">` koristi se HTML-element `textarea`.

Ovo su osnovni atributi koji se koriste kod elementa `textarea`:

- **name** – naziv polja koji se šalje zajedno s unesenim podacima
- **maxlength** – maksimalni dozvoljeni broj znakova
- **placeholder** – tekst koji daje primjer unosa

Primjer:

```
<textarea name="komentar" maxlength="500"></textarea>
```



Ovaj element uvijek mora imati početnu i završnu oznaku. Ne može se pisati samo `<textarea>`, već je uvijek potrebno napisati `<textarea></textarea>`.

Ako je potrebno napisati početnu vrijednost koja će biti prikazana unutar ovoga polja, nju je potrebno ispisati unutar oznaka `<textarea>` .

Unutar ovoga elementa korisnik može prelamati tekst u više redova.

11.15. Označavanje elemenata obrasca (*label* element)

Uz svako polje u obrascu potrebno je navesti i njegov naziv tj. oznaku. Taj tekst daje značenje pojedinom polju i poželjno ga je staviti unutar HTML-elementa `label`.

Osim vizualno, element `label` treba i u kôdu povezati s odgovarajućim elementom polja, kako bi se olakšala upotreba obrasca osobama koje koriste čitače ekrana (kako bi čitač znao pročitati ispravnu oznaku polja).

Ovo povezivanje postiže se tako da se na element *label* postavi atribut *for* s istom vrijednošću koju ima atribut *id* na odgovarajućem elementu *input* (ili drugom elementu obrasca).

Primjer:

```
<label for="ime">Ime i prezime:</label>
<input type="text"id="ime"name="ime">
```



Dodatna prednost jest što se klikom na oznaku postavlja fokus na polje za unos teksta. Kod dugmadi za odabir ili kvačica za označavanja ovo ponašanje je još korisnije – klikom na oznaku (element <label>) automatski se odabire dugme (odnosno kvačicu), čime postaje lakše odabrati željenu opciju (područje koje je osjetljivo na klik je znatno veće).

Primjer:

```
<input type="radio" id="musko" name="spol" value="m">
<label for="musko">muško</label>

<input type="radio" id="zensko" name="spol"
value="z">
<label for="zensko">žensko</label>

<input type="radio" id="ostalo" name="spol"
value="n">
<label for="ostalo">ne želim odgovoriti</label>
```



11.16. Grupiranje elemenata obrasca

Kako bi obrasci koji sadrže puno polja bili pregledniji, potrebno je vizualno grupirati srodne elemente u odvojene cjeline.

Takve cjeline dobivaju se upotrebom elementa `fieldset`, unutar kojega se navode elementi obrasca koji pripadaju toj cjelini.

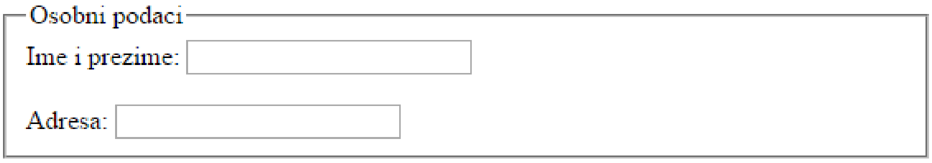
Kao prvi element unutar elementa `fieldset` može doći element `legend`, unutar kojega se navodi naslov pojedine cjeline.

Primjer:

```
<fieldset>
      <legend>Osobni podaci</legend>
      <label for="ime">Ime i prezime:</label>
      <input id="ime" type="text" name="ime">
      <br><br>
      <label for="adresa">Adresa:</label>
      <input type="text" id="adresa" name="adresa">
</fieldset>
```

Zanimljivosti i napomene

Iako su na prvi pogled slični, atributi `id` i `name` imaju različitu namjenu. Atribut id služi za referenciranje elementa iz *JavaScripta* i CSS-a (te za povezivanje s elementom label)te mora biti jedinstven unutar dokumenta.
Atribut `name` služi da bi se postavio naziv koji označava poslani podatak, i više elemenata (npr. grupa gumba za odabir) može imati isti `name`.



11.17. Validacija podataka unesenih u obrazac

Zanimljivosti i napomene

Podatke dobivene iz HTML obrasca, svakako je potrebno provjeriti prije spremanja u bazu, bez obzira na eventualne provjere u samome HTML obrascu.

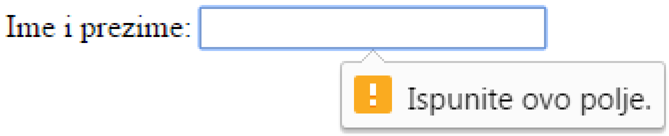
Kako bi se onemogućilo slanje neispravnih podataka, u preglednik su ugrađene određene mogućnosti provjere, odnosno validacije upisanih podataka. Već spomenuti atribut `maxlength` ograničava duljinu upisanoga teksta, te je moguća provjera i ograničenje unosa kod specijalnih tipova podataka (brojčani te datumski podatak, URL i e-mail adresa).

Glavni i najčešći način validacije polja je provjera je li ono popunjeno (ako se radi o obaveznom polju). To se postiže pomoću atributa `required` koji je moguće postaviti na većini elemenata obrasca.

Polje na kojem je postavljen ovaj atribut mora biti popunjeno kako bi se obrazac mogao poslati poslužitelju.

Primjer:

```
<input type="text" name="ime" required>
```



12. Uključivanje vanjskih stranica

Po završetku ovog poglavlja polaznik će moći:

- *kao dio stranice uključiti tuđu web-stranicu ili dio web-stranice koristeći element <iframe>.*

HTML omogućuje da se kao dio *web*-stranice uključi neka tuđa *web*-stranica ili dio iste. To bi se moglo opisati i kao "prozor" koji gleda na drugu stranicu.

Jedan od uobičajenih načina na koji se vanjske stranice uključuju je uključivanje Google mape na stranicu. Uključiti se može bilo koja stranica, bilo s vlastitoga *web*-sjedišta ili tuđega, koje se nalazi na drugom poslužitelju.

12.1. Uključivanje vanjske stranice – *iframe* element

HTML-element koji se koristi za uključivanje vanjske stranice je `<iframe>`. Ovaj element može imati sljedeće atribute:

- `src` – atribut je kojemu se dodjeljuje URL te stranice koja će se prikazati u okviru
- `height` – određuje visinu okvira u pikselima
- `width` – određuje širinu okvira u pikselima
- `seamless` – omogućava korisniku kretanje unutar sadržaja u `<iframe>` elementu u slučaju da je on veći od okvira.

Primjer uključivanja vanjske stranice (Google mape):

```
<iframe width="425" height="350"
src="https://maps.google.com/maps/ms?msid=20024248206
0725049899.00048c0cc1718c9a125c0&amp;msa=0&amp;ie=UTF
8&amp;ll=45.794639,15.956783&amp;spn=0.054458,0.08540
2&amp;t=m&amp;output=embed">

</iframe>
```

Zanimljivosti i napomene

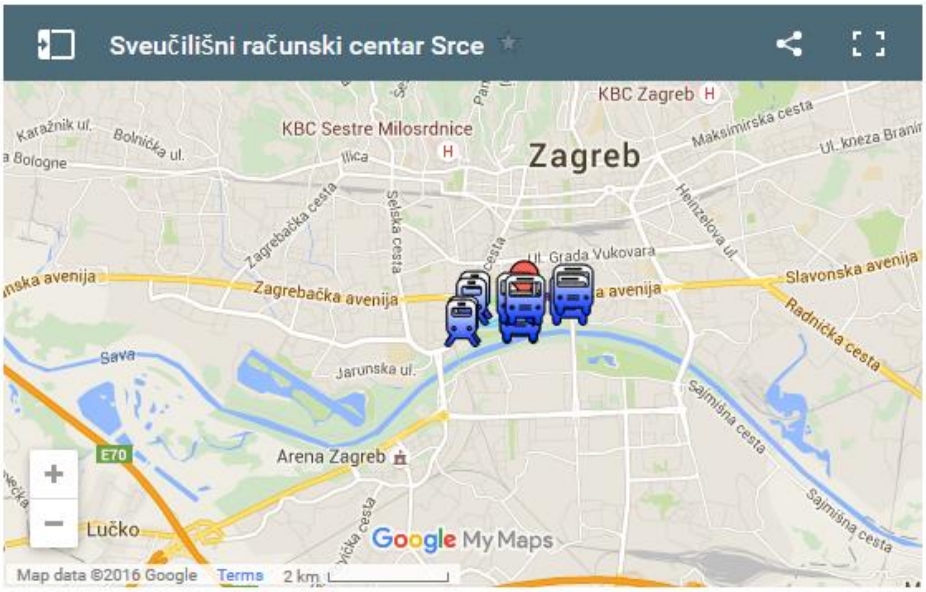
Embeddanjem se u vlastiti HTML-dokument uključuje dio tuđega HTMLkoda, dok se uz pomoć `<iframe>` elementa uključuje dio tuđe web stranice.

Zanimljivosti i napomene

HTML5 više ne podržava atribute `scrolling` i `frameborder`.

`Scrolling` je atribut koji određuje hoće li okvir imati klizne trake u slučaju da je sadržaj veći od okvira ili ne. Može imati vrijednosti „da“, „ne“ ili „auto“ u slučaju da se trake pojavljuju automatski kada je to potrebno.

`Frameborder` je atribut koji određuje hoće li <`iframe`> imati okvir ili ne. Može imati vrijednosti 0 ili 1.


Sveučilišni računski centar Srce
KBC Zagreb
KBC Sestre Milosrdnice
Zagreb
Maksimirska cesta
Ilica
Selska cesta
Zagrebačka cesta
Zagrebačka avenija
Ul. Grada Vukovara
Slavonska avenija
Radnička cesta
Sava
Jarunska ul.
Arena Zagreb
E70
Lučko
Sajmišna cesta
Google My Maps
Map data ©2016 Google
Terms
2 km

12.2. Vježba: Obrazac i uključivanje vanjske stranice

Obrazac

1. Otvorite postojeći HTML-dokument **prijava.html** u mapi **C201/vjezbe/vjezba5**.
2. Nakon uvodnih odlomaka umetnite element `form` koji će predstavljati obrazac za prijavu na tečajeve Srca. Kao način slanja podataka postavite POST, a kao URL na koji se šalju podaci postavite adresu http://www.htmlcodetutorial.com/cgi-bin/mycgi.pl (radi se o pokaznoj stranici na Internetu koja će ispisati sve podatke poslane putem obrasca.)
3. Neka se obrazac sastoji od dva dijela (elementa `fieldset`) – "Osobni podaci" i "Odabir tečaja i ostalo".
4. U prvi dio obrasca dodajte polja za unos imena i prezimena. Za opis svakoga polja u obrascu koristite element `<label>`.
5. U prvi dio obrasca dodajte polje za unos e-mail adrese.
6. U prvi dio obrasca dodajte *label* element s tekstom "Status polaznika" te zatim grupu dugmadi za odabir pomoću kojih će biti moguće odabrati status – *student*, *član ak. zajednice* ili *ostali.* Neka vrijednost *student* bude predodabrana. Tekst uz pojedine gumbe za odabir neka također bude unutar elemenata *label*, kako bi klik na tekst označio odgovarajući gumb.
7. U drugi dio obrasca dodajte listu za odabir tečaja. Ponuđene tečajeve možete kopirati sa stranice "Popis tečajeva".
8. Dodajte polje za unos prijedloga.
9. Dodajte kvačicu za označavanje koju opisuje tekst "Da, želim primati obavijesti" (unutar povezanog elementa `label`).
10. Postavite kao obavezna za unos sljedeća polja: ime, prezime, e-mail adresu i odabir tečaja.
11. Postavite maksimalnu duljinu teksta koju je moguće unijeti u polja za ime (50 znakova), prezime (50 znakova), e-mail (100 znakova) i komentar (250 znakova).
12. Na kraju obrasca dodajte gumb za slanje podataka.
13. Isprobajte unos i slanje podataka putem ovog obrasca.

Uključivanje vanjskih stranica

14. Otvorite postojeći HTML-dokument **kontakt.html** u mapi **C201/vjezbe/vjezba5**.
15. U njega umetnite element *iframe* koji uključuje *GoogleMaps* kartu s lokacijom Sveučilišnog računskog centra.

Bilješke: